

E-1 DYNAMIC FLUID-FLOW MODEL UPDATE

EASY/ROCETS Enhancement and
Model Development Support

Final Report

July 1, 1998

Contract Number:
NAS13-564
Delivery Order 143

Submitted to:

NASA John C. Stennis Space Center
Stennis Space Center, MS 39529

Randolph F. Follett
Electrical and Computer Engineering Department
and
Robert P. Taylor
Mechanical Engineering Department
Mississippi State University
Mississippi State, MS 39762

PREFACE

This report documents the research conducted to update computer models for dynamic fluid flow simulation of the E-1 Test Stand subsystems at the NASA John C. Stennis Space Center. This work also involved significant upgrades to the capabilities of the EASY/ROCETS library through the inclusion of the NIST-12 thermodynamic property database and development of new control system modules. This work was funded under contract NAS13-564, delivery order 143.

The authors thank Mr. Randy Holland, Mr. Larry deQuay, and Mr. Bud Nail of NASA and Mr. Steve Poulton of Lockheed-Martin for their support and encouragement during the course of this research.

ABSTRACT

Computer models for dynamic fluid flow simulation are needed to predict the pressures, temperatures, flow rates, etc., in present and future testing operations at the NASA John C. Stennis Space Center (SSC). Such simulations are used in facility design, test scenario development, facility modification, and facility control. The ROCKET Engine Transient Simulation (ROCETS) package, which was initially developed by Pratt and Whitney for NASA Marshall Space Flight Center, and EASY5x, which is a commercial package developed by the Boeing Co., are the two major components which comprise the EASY/ROCETS dynamic fluid flow simulation package which has been developed by Mississippi State University (MSU) personnel for use by NASA/SSC. Additional code has been written to handle tasks specific to ground-test facility modeling such as gas bottles and pressurized liquid run tanks.

In the present research, incorporating the complete NIST-12 property database into the EASY/ROCETS library has significantly enhanced the thermodynamic property tables. Previously models for the low-pressure and high-pressure run systems for the E-1 Test Stand were provided. These models have been updated based on the information provided by Mr. Larry deQuay. In addition, new modules have been developed to model the Allen-Bradley PID programmable controllers and for a valve with a time delay input.

Items for continued development are outlined in the report.

TABLE OF CONTENTS

Section	Title	Page
	PREFACE.....	i
	ABSTRACT.....	ii
	TABLE OF CONTENTS.....	iii
1	INTRODUCTION.....	1
2	THERMODYNAMIC PROPERTY TABLE ENHANCEMENT AND DEVELOPMENT.....	3
3	E-1 MODEL DEVELOPMENT UPDATES.....	66
4	NEW MODEL DEVELOPMENT.....	96
5	CONCLUSIONS AND RECOMMENDED FUTURE WORK.....	115
6	REFERENCES.....	117
	APPENDIX I.....	118

SECTION 1

INTRODUCTION

Computer models for dynamic fluid flow simulation are needed to predict the pressures, temperatures, flow rates, etc., in present and future testing operations at the NASA John C. Stennis Space Center (SSC). Such simulations are used in facility design, test scenario development, facility modification, and facility control. The ROcket Engine Transient Simulation (ROCETS) package, which was initially developed by Pratt and Whitney for NASA Marshall Space Flight Center, and EASY5x, which is a commercial package developed by the Boeing Co., are the two major components which comprise the EASY/ROCETS dynamic fluid flow simulation package which has been developed by Mississippi State University (MSU) personnel for use by NASA/SSC. Additional code has been written (Taylor and Follett, 1994, and Follett and Taylor, 1996) to handle tasks specific to ground-test facility modeling such as gas bottles and pressurized liquid run tanks.

In the present research, incorporating the complete NIST-12 property database into the EASY/ROCETS library has significantly enhanced the thermodynamic property tables. These revisions are discussed in Section 2.

Previously models for the low-pressure and high-pressure run systems for the E-1 Test Stand (Follett and Taylor, 1996) were provided. These models have been updated based on the information provided by Mr. Larry deQuay (1997). These new models are presented in Section 3.

In addition new modules have been developed to model the Allen-Bradley PID programmable controllers and for a valve with a time delay input. Section 4 presents the

development of these new modules. Conclusions and recommendations for future work are presented in Section 5. Finally the notes for model development are presented in the Appendix.

SECTION 2

THERMODYNAMIC PROPERTY TABLE ENHANCEMENT AND DEVELOPMENT

The thermodynamic property tables that were programmed with the original ROCETS package were meant for rocket engine simulation and, hence, had abridged versions of the property tables for N₂ and He. Furthermore, no properties were supplied for Air. As need has arisen, property tables have been added in an ad hoc manner for Air and N₂. A primary motivation for this work was the failure of the existing package with high pressure (14,000 psi) N₂ needed to model the high-pressure O₂ system on the E-1 Test Stand (formerly known as the CTF).

After review of the alternatives, it was decided that the best way to proceed was to incorporate the NIST-12 (1995) property FORTRAN codes directly into the EASY/ROCETS library. This was done in two stages. First, the NIST-12 routines were programmed into the existing EASY/ROCETS Library, "er," for the N₂ properties. This allows existing models to be run in the updated system without rebuilding the models and with only slight modification. Next the entire EASY/ROCETS library was redesigned from the ground up to use the NIST-12 routines for all property computations. The 1995 version of the NIST-12 property routines would not handle mixed-phase states and performed poorly for thermodynamic states that were near the mixed-phase region—near the "dome." In the course of this work, the NIST-12 codes were modified to compute the thermodynamic properties of mixed-phase states. Furthermore, an error was discovered in the 1995 version of the FORTRAN programs. Correction of this error allows the revised code to compute properties near the dome with better accuracy and more reliably.

“er” Library Modifications

A major problem with the original ROCETS package was the abridged nature of the N2 property tables. When the high-pressure O2 system on the E-1 Test Stand (formerly know as CTF) was first modeled using EASY/ROCETS (Follett and Taylor, 1996), the model results were inaccurate because of the limits of the N2 property tables. As a first step, the existing EASY/ROCETS library (“er” in the EASY/ROCETS library list) was modified to use the NIST-12 property routines (corrected-expanded version discussed later) for the N2 property calculations. The other property tables, H2 and O2, were not modified. This will allow existing models to be used with high-pressure N2 requirements; however, it is not the most efficient way to use the NIST-12 codes and does not make use of all the additional capabilities that the routines make available. A completely redesigned library is discussed in a following section.

Table 1 shows the modified version of the “er” library subroutine N2PROP. Modifications are highlighted in bold italic type. Review of the table reveals that all options except “ $P = F(\text{RHO}, H)$ ” are computed from the NIST-12 routines. “ $P = F(\text{RHO}, H)$ ” was omitted since it is not currently used in EASY/ROCETS.

Figure 1 shows a test model that was used to demonstrate that the modified library can handle high-pressure N2. The model is very simple with a bottle blowing down through a valve against a backpressure in the exit module. Figure 2 shows the bottle pressure history during the blow down. The figure shows that the pressure history is smooth and continuous not stair-step as with the unmodified library (Follett and Taylor, 1996).

To demonstrate the mixed-phase capabilities of the modified library, the test model in Figure 3 was built and executed. The inlet feeds saturated N2 vapor at 50 psia through the first valve into a volume that initially contains a saturated mixture of N2 vapor and liquid at 50 psia. This volume discharges through a valve against a backpressure of 25 psia in the exit. Figure 4 shows the pressure and temperature history of the volume, "V1." As the fluid in the volume discharges through the valve, the pressure in the volume reduces. As this pressure reduces, the inflow of saturated vapor from the inlet increases. Looking at the temperature plot, it is seen that the temperature in the volume first decreases following the saturation temperature line as the pressure decreases. At about 50 sec, all of the original mixed-phase fluid has been discharged or boiled off. After this time, the temperature increases as the fluid in the volume is replaced by the warmer vapor from the inlet. Figure 5 shows the computed flow rates.

This model assumes that the mixed-phase is continuously mixed (as a mist, for example) not separated and that the flow of mixed phase has no effect on the flow properties of the valve.

Figure 6 shows the old model (Follett and Taylor, 1996) for the high-pressure O2 system on the E-1 Test Stand. The need for high-pressure N2 in the gas pressurization subsystem was one of the driving forces behind the present work to upgrade the property computations in EASY/ROCETS. Figure 7 shows the runtank and N2 bottle pressure histories. The runtank pressure controller adjusts the gas control valve to maintain the runtank pressure at a constant level in this case. As the bottle pressure decreases, the gas control valve must open up more to maintain the constant ullage pressure. Figure 8 shows the necessary control valve Cv values. Figure 9 shows a comparison of the

computed value of the liquid flow rate and the desired flow rate as computed by Larry deQuay (1996).

Since we were modifying the “er” library, we took the opportunity to update the input for the runtank module. Figure 10 shows the input window. The initial conditions are now set by inputting the ullage vapor enthalpy and pressure, HVG and PVG, the liquid enthalpy, HLG, and the vapor volume of the ullage, VVG, in the “Inputs” list on the left of the window. This change makes the runtank input consistent with the other modules in the “er” library. All initial inputs come as pressure and enthalpy.

Table 1. Modified N2 Property Subroutine for er Library Using NIST-12 Routines.

```

C      DATA SET MSFCN2PROP AT LEVEL 001 AS OF 08/09/90
C      SUBROUTINE N2PROP (IUPDAT , MODLOC , OPTION , IDSC , XVAR ,
C      *                  YVAR , OVAR , PNTX , PNTY , IDSCR )
C*****
C %BEGIN CLASS N2PROP
C
C      SUBPROGRAM N2PROP          UNCLASSIFIED          SID: E950
C
C      UNITED TECHNOLOGIES CORPORATION
C      PRATT & WHITNEY
C      WEST PALM BEACH, FLORIDA
C %END CLASS N2PROP
C*****
C %BEGIN PURPOSE N2PROP
C
C      THIS SUBROUTINE ACCESSES NITROGEN PROPERTIES
C      VIA NIST_12 NITROGEN PROPERTY Programs.
C %END PURPOSE N2PROP
C*****
C %BEGIN HISTORY N2PROP
C
C      WRITTEN 11/13/89 BY J. P. SPINN
C
C      NIST 12 CODE ADDED 1998 BY BOB TAYLOR MISS. STATE U.
C      FOR EASY/ROCEFS IMPLEMENTATION
C %END HISTORY N2PROP
C*****
C %BEGIN DESCRIPTION N2PROP
C
C      INPUTS:
C
C      IUPDAT = TRANSIENT UPDAT FLAG
C      MODLOC = 4 CHARACTER PROPERTY NODE LOCATION
C      OPTION = 12 CHARACTER READ OPTION
C      IDSC   = DISCRETE INPUT INDICATING MAP RANGE
C      XVAR   = FIRST INDEPENDENT PARAMETER
C      YVAR   = SECOND INDEPENDENT PARAMETER (MUST BE MAP FAMILY)
C      OVAR   = TEMPERATURE GUESS VALUE (ALSO OUTPUT VALUE OF PROPERTY)
C
C      OUTPUT/INPUT:
C
C      PNTX   = X POINT NUMBER
C      PNTY   = Y POINT NUMBER
C
C      OUTPUT:
C
C      OVAR   = DEPENDENT PARAMETER
C      IDSCR  = DISCRETE REQUEST INDICATING MAP REGION
C
C %END DESCRIPTION N2PROP
C*****
C %BEGIN COMMENTS N2PROP
C
C      1.  THE OPTION INPUT IS INTENDED TO REPRESENT THE FUNCTIONAL
C          RELATIONSHIP DESIRED.  THE OPTION CODE, INDEPENDENT
C          THERMODYNAMIC STATES, DEPENDENT STATE, AND MAP SUBROUTINE
C          CALLED ARE SHOWN IN THE FOLLOWING TABLE.  NOTE THAT THE

```

Table 1. Modified N2 Property Subroutine for er Library Using NIST-12 Routines.

C	FORMAT OF THE OPTION ALWAYS HAS THE MAP FAMILY PARAMETER						*
C	AS THE SECOND INDEPENDENT PARAMETER.						*
C							*
C	-----						*
C	OPTION	XVAR	YVAR	OVAR	MAP		*
C	-----						*
C	P=F(RHO,U)	DENSITY	INTER ENG	PRESSURE	ITERAT		*
C	P=F(RHO,H)	DENSITY	ENTHALPY	PRESSURE	NRHP05		*
C	RHO=F(P,H)	PRESSURE	ENTHALPY	DENSITY	NRHP05		*
C	T=F(P,H)	PRESSURE	ENTHALPY	TEMPERATURE	NPHT05		*
C	S=F(H,P)	ENTHALPY	PRESSURE	ENTROPY	NHPS05		*
C	H=F(S,P)	ENTROPY	PRESSURE	ENTHALPY	NHPS05		*
C	CP=F(H,P)	ENTHALPY	PRESSURE	SPEC HEAT	FLUIDS		*
C	CV=F(H,P)	ENTHALPY	PRESSURE	SPEC HEAT	FLUIDS		*
C	GAMA=F(H,P)	ENTHALPY	PRESSURE	RATIO SP	FLUIDS		*
C	MU=F(H,P)	ENTHALPY	PRESSURE	VISCOSITY	FLUIDS		*
C	K=F(H,P)	ENTHALPY	PRESSURE	THERMAL COND	FLUIDS		*
C	-----						*
C							*
C							*
C	%END COMMENTS N2PROP						*
C	*****						*
C	%BEGIN INTERFACE N2PROP						*
C							*
C	-----						*
C	CALL LIST	SYSTEM	SYSTEM TAG	ARRAY	I/O	VAR	*
C	NAME	NAME		STATUS	STATUS	TYPE	*
C	-----						*
C	IUPDAT	IUPDAT	GLOBAL		IN 0	I*4	*
C	MODLOC	MODL	NAME 0		IN 0	C*4	*
C	OPTION	OPTN	VARIABLE		IN 0	C*12	*
C	IDSC	DSC	DSC		IN 0	I*4	*
C	XVAR	XVAR	VARIABLE		IN 0	R*4	*
C	YVAR	YVAR	VARIABLE		IN 0	R*4	*
C	OVAR	OVAR	VARIABLE		OUT 0	R*4	*
C	PNTX	PNTX	VARIABLE		OUT 0	R*4	*
C	PNTY	PNTY	VARIABLE		OUT 0	R*4	*
C	IDSCR	DSCR	DSCR		OUT 0	I*4	*
C	-----						*
C							*
C	%END INTERFACE H2PROP						*
C	*****						*
C	%BEGIN SUBROUTINES REQUIRED N2PROP						*
C							*
C	SUBROUTINES REQUIRED : NPHT05, NRHP05, NHPS05, ERCK00, NIST_12						*
C							*
C	%END SUBROUTINES REQUIRED N2PROP						*
C	*****						*
C	%BEGIN COMMONS REQUIRED N2PROP						*
C							*
C	COMMONS REQUIRED: GUNITS,						*
C							*
C	%END COMMONS REQUIRED N2PROP						*
C	*****						*
	implicit real*8(a-h,o-z)						
	CHARACTER*12 OPTION						
	CHARACTER*4 MODLOC						
	COMMON / GUNITS / IUNIT , GC , GR , RJ , RU ,						
	* CLEN , CMASS , CFORCE , CTEMP , CENERGY,						
	* FLOCON						
C	*****						*
	IF (IUNIT.EQ.0) THEN						
	CFOOT=12.						

Table 1. Modified N2 Property Subroutine for er Library Using NIST-12 Routines.

```

      CVIS=12.
      CTHK=12*3600.
    ELSE
      CFOOT=1.
      CVIS=1000.
      CTHK=1.
    END IF
    T = OVAR
C
C*****
C
    IF(      INDEX(OPTION,  'P=F(RHO,U)'      ) .NE. 0) THEN
      D = XVAR*CFOOT**3
      U = YVAR
      IFL = 2
      IOP = 5
      CALL FPROP(IFL,IOP,IUNIT,P,T,QL,D,U,H,S,CP,CV,V,TH,SO,DI)
      OVAR = P
C
C*****
C
    ELSEIF(      INDEX(OPTION,  'P=F(RHO,H)'      ) .NE. 0) THEN
      IOPT = 1
      RHO = XVAR
      H = YVAR
      CALL NRHP05 (IUPDAT , MODLOC , IOPT , IDSC , RHO ,
*                H , P , PNTX , PNTY , IDSCR )
      OVAR = P
C
C*****
C
    ELSEIF( INDEX(OPTION,  'RHO=F(P,H)'      ) .NE. 0) THEN
      IFL = 2
      IOP = 6
      H = YVAR
      P = XVAR
      CALL FPROP(IFL,IOP,IUNIT,P,T,QL,D,U,H,S,CP,CV,V,TH,SO,DI)
      OVAR = D/(CFOOT**3)
C
C*****
C
    ELSEIF( INDEX(OPTION,  'S=F(H,P)'      ) .NE. 0) THEN
      IFL = 2
      IOP = 6
      H = XVAR
      P = YVAR
      CALL FPROP(IFL,IOP,IUNIT,P,T,QL,D,U,H,S,CP,CV,V,TH,SO,DI)
      OVAR = S
C
C*****
C
    ELSEIF( INDEX(OPTION,  'H=F(S,P)'      ) .NE. 0) THEN
      IOP = 4
      IFL = 2
      S = XVAR
      P = YVAR
      CALL FPROP(IFL,IOP,IUNIT,P,T,D,U,H,S,CP,CV,V,TH,SO,DI)
      OVAR = H
C
C*****
C
    ELSEIF( INDEX(OPTION,  'T=F(P,H)'      ) .NE. 0) THEN

```

Table 1. Modified N2 Property Subroutine for er Library Using NIST-12 Routines.

```

      IFL = 2
      IOP = 6
      P   = XVAR
      H   = YVAR
      CALL FPROP (IFL, IOP, IUNIT, P, T, QL, D, U, H, S, CP, CV, V, TH, SO, DI)
      OVAR = T
C
C*****
C
      ELSEIF ( INDEX (OPTION, 'F(H,P)' ) .NE. 0) THEN
        IFL = 2
        IOP = 6
        P   = YVAR
        H   = XVAR
        CALL FPROP (IFL, IOP, IUNIT, P, T, QL, D, U, H, S, CP, CV, V, TH, SO, DI)
C
C*****
C
      IF (INDEX (OPTION, 'CP=' ) .NE. 0) THEN
        OVAR = CP
C
C*****
C
      ELSE IF (INDEX (OPTION, 'CV=' ) .NE. 0) THEN
        OVAR = CV
C
C*****
C
      ELSE IF (INDEX (OPTION, 'GAMA=' ) .NE. 0) THEN
        GAMA = CP / CV
        OVAR = GAMA
C
C*****
C
      ELSE IF (INDEX (OPTION, 'MU=' ) .NE. 0) THEN
        RMU = V / CVIS
        OVAR = RMU
C
C*****
C
      ELSE IF (INDEX (OPTION, 'K=' ) .NE. 0) THEN
        RK = TH / CTHK
        OVAR = RK
C
C*****
C
      ELSE
        CALL ERCK00 (IUPDAT, 'N2PROP ', MODLOC// ' 1st', 10000,
*          '--- INVALID OPTION IN N2PROP ---' )
        RETURN
      END IF

      ELSE
        CALL ERCK00 (IUPDAT, 'N2PROP ', MODLOC// ' 2nd', 10000,
*          '--- INVALID OPTION IN N2PROP ---' )
        GOTO 99
      ENDIF
        CALL ERCK00 (IUPDAT, 'N2PROP ', MODLOC// ' 3rd', 0000,
*          '--- VALID OPTION IN N2PROP ---' )
C
1000 CALL ERCK00 (IUPDAT, 'N2PROP ', MODLOC// ' 4th', 0000,
*          '--- PRESSURE NOT FOUND IN N2PROP ---' )

```

Table 1. Modified N2 Property Subroutine for er Library Using NIST-12 Routines.

```

      GOTO 99
2000 CALL ERCK00 (IUPDAT , 'N2PROP ', MODLOC// ' 5th', 0000 ,
*          '--- ITERATIONS DONOT CONVERGE IN N2PROP --- ')
      GOTO 99

99  CONTINUE
    RETURN
    END
C      DATA SET MSFCERCK00 AT LEVEL 001 AS OF 08/09/90

```

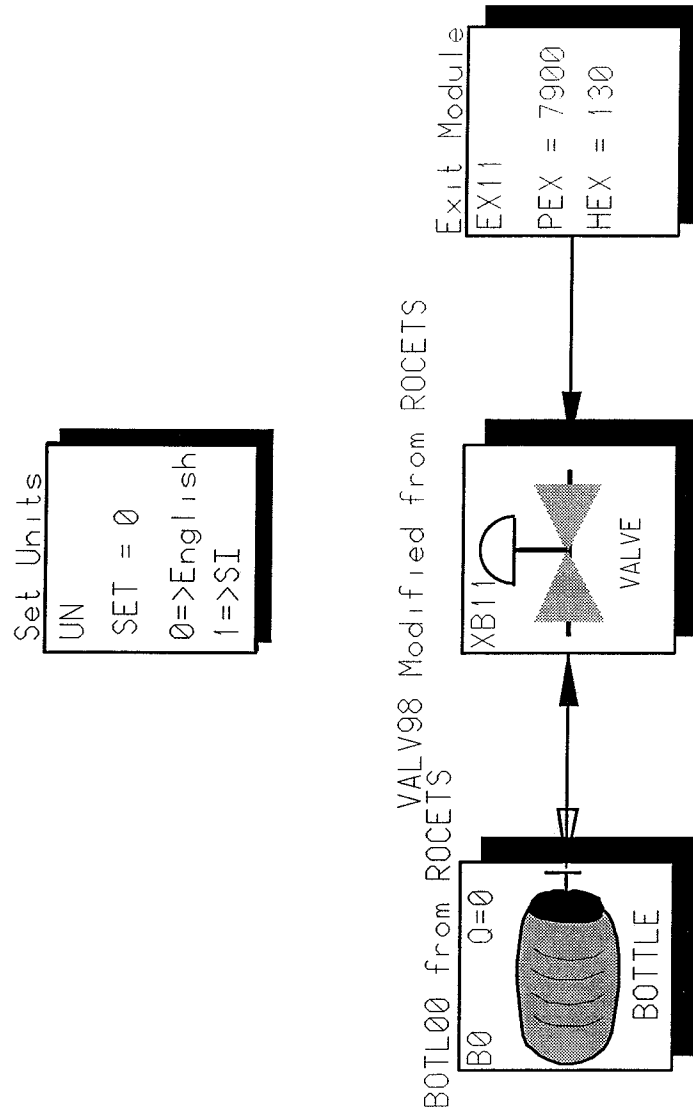


Fig. 1. High-Pressure N2 Property Test Model

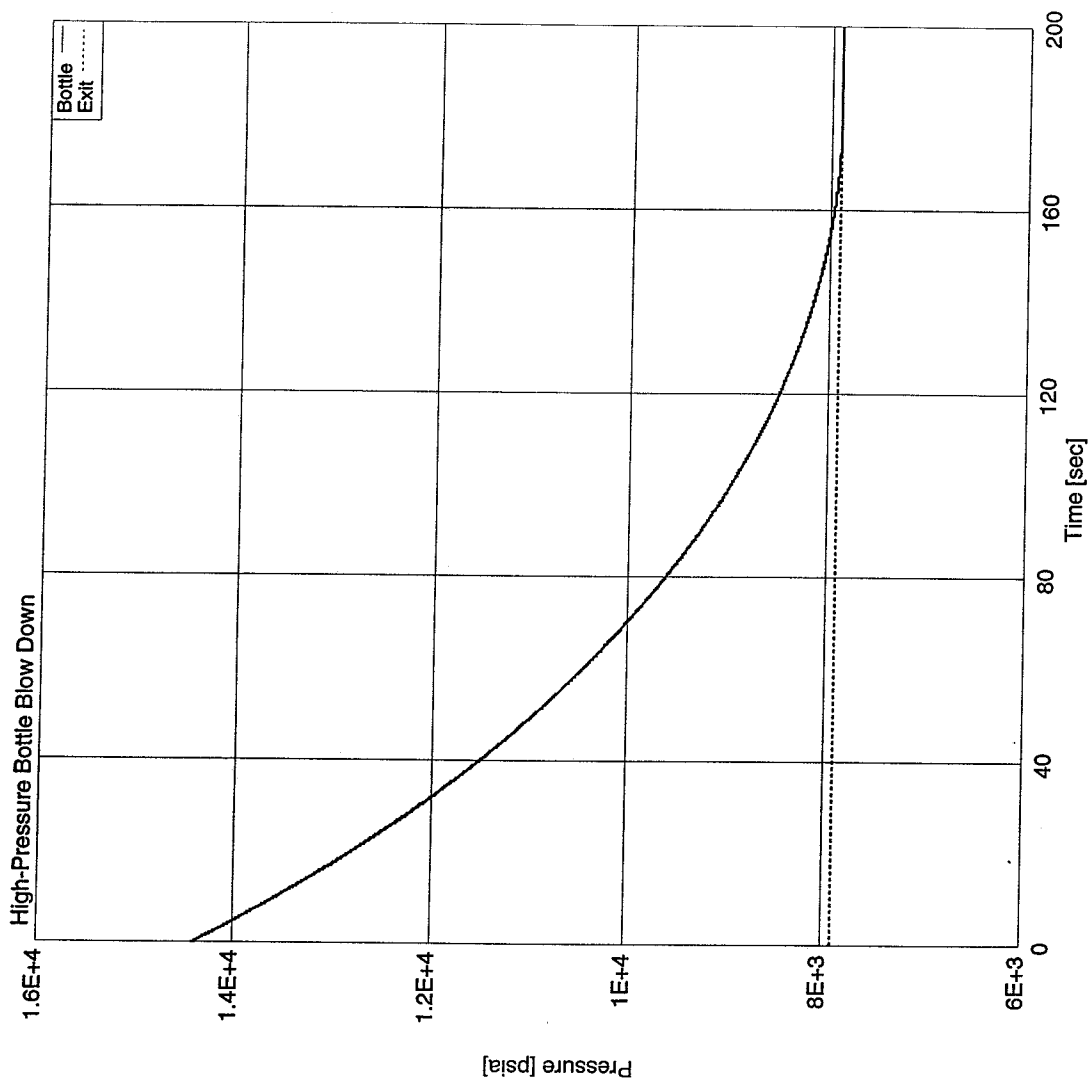


Fig. 2. Test Model Pressure History.

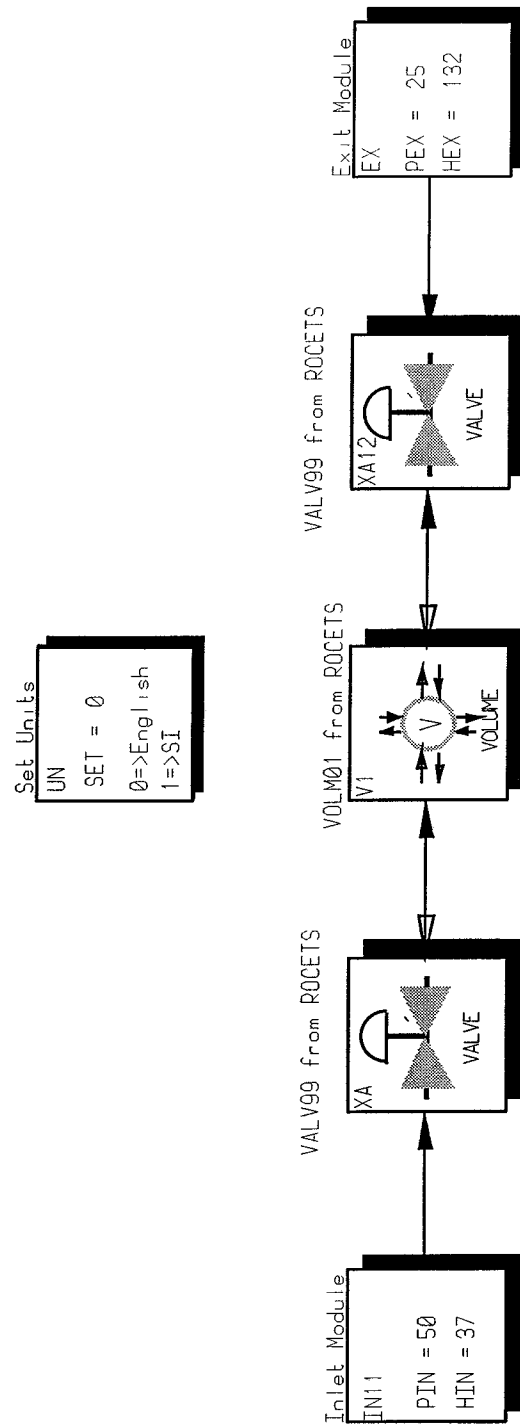


Fig. 3. Mixed-Phase Test Model

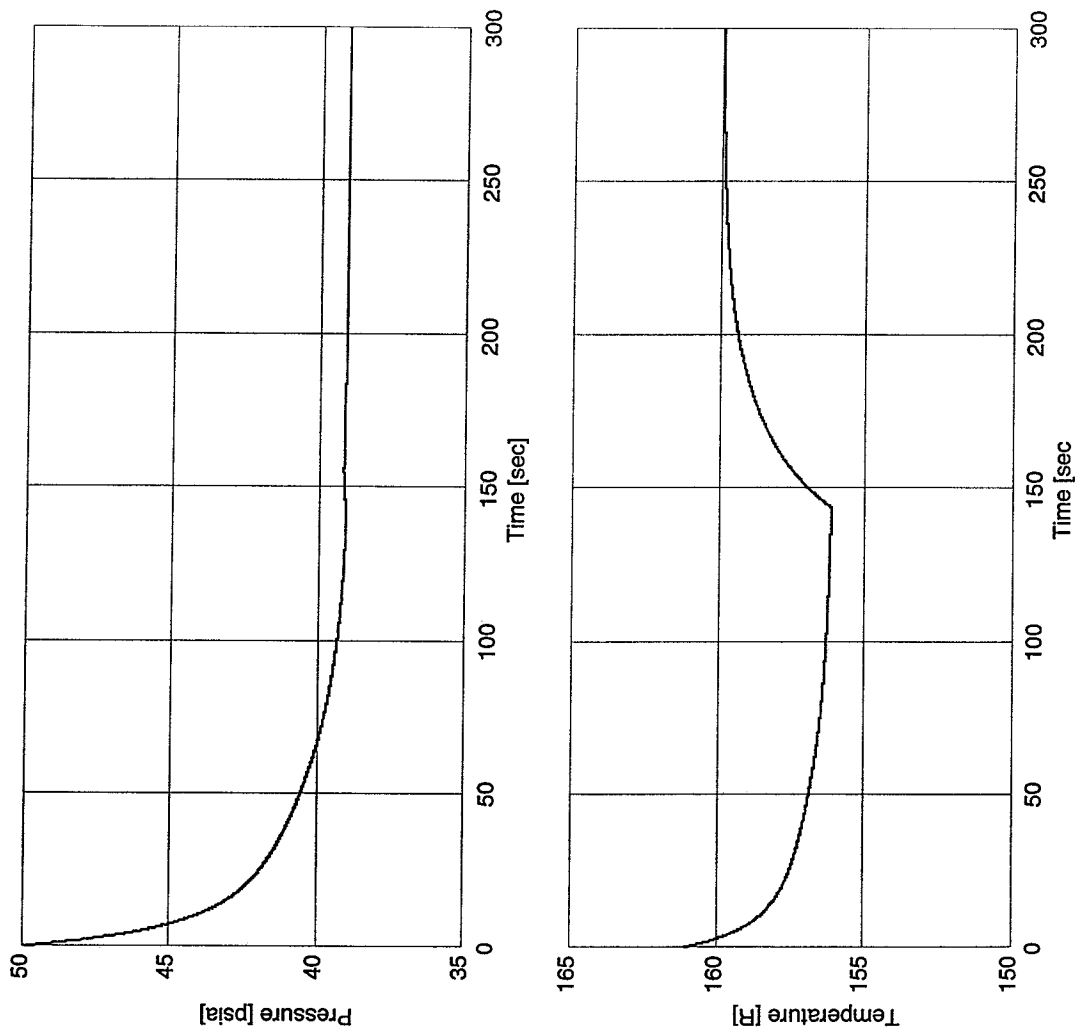


Fig. 4. Mixed-Phase Test Model Pressure and Temperature Histories.

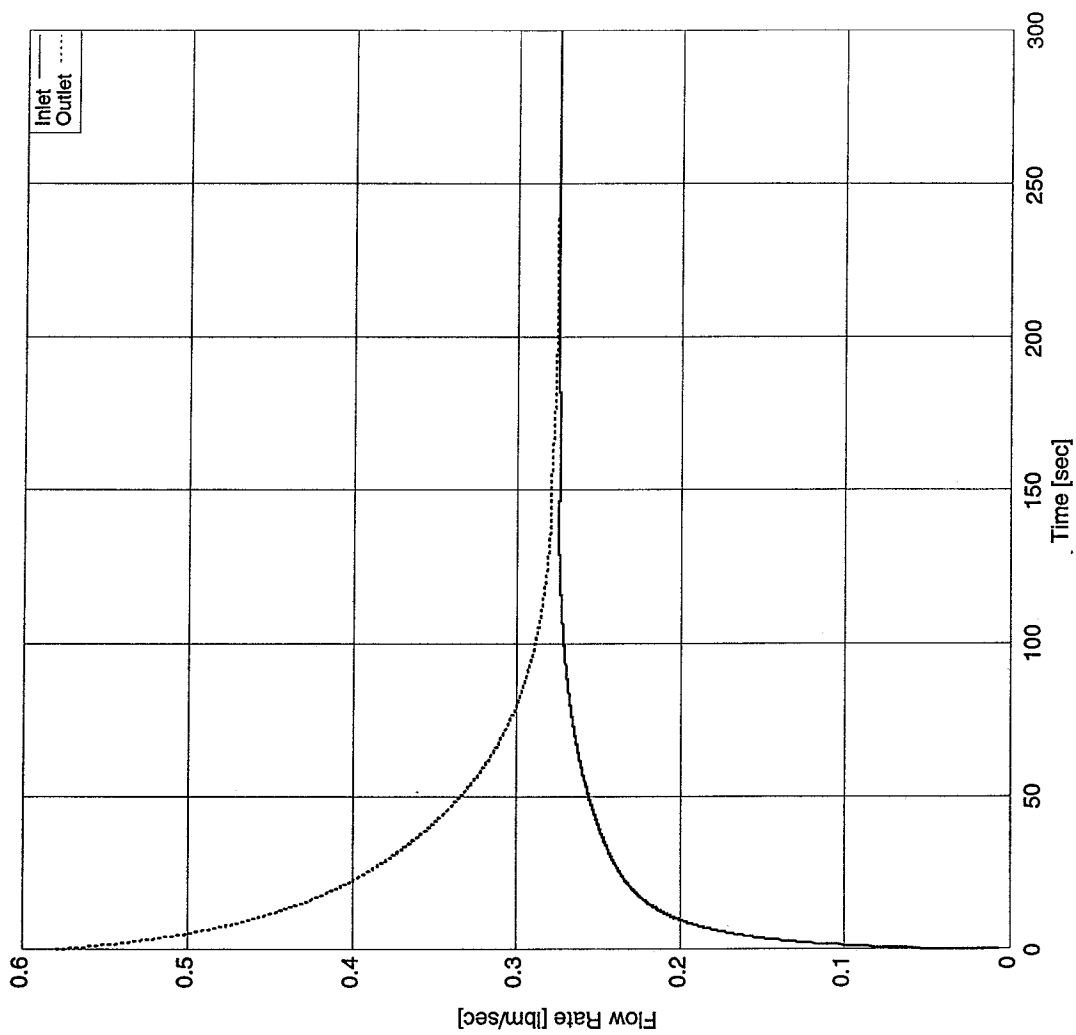


Fig. 5. Mixed-Phase Test Model Flow Rates.

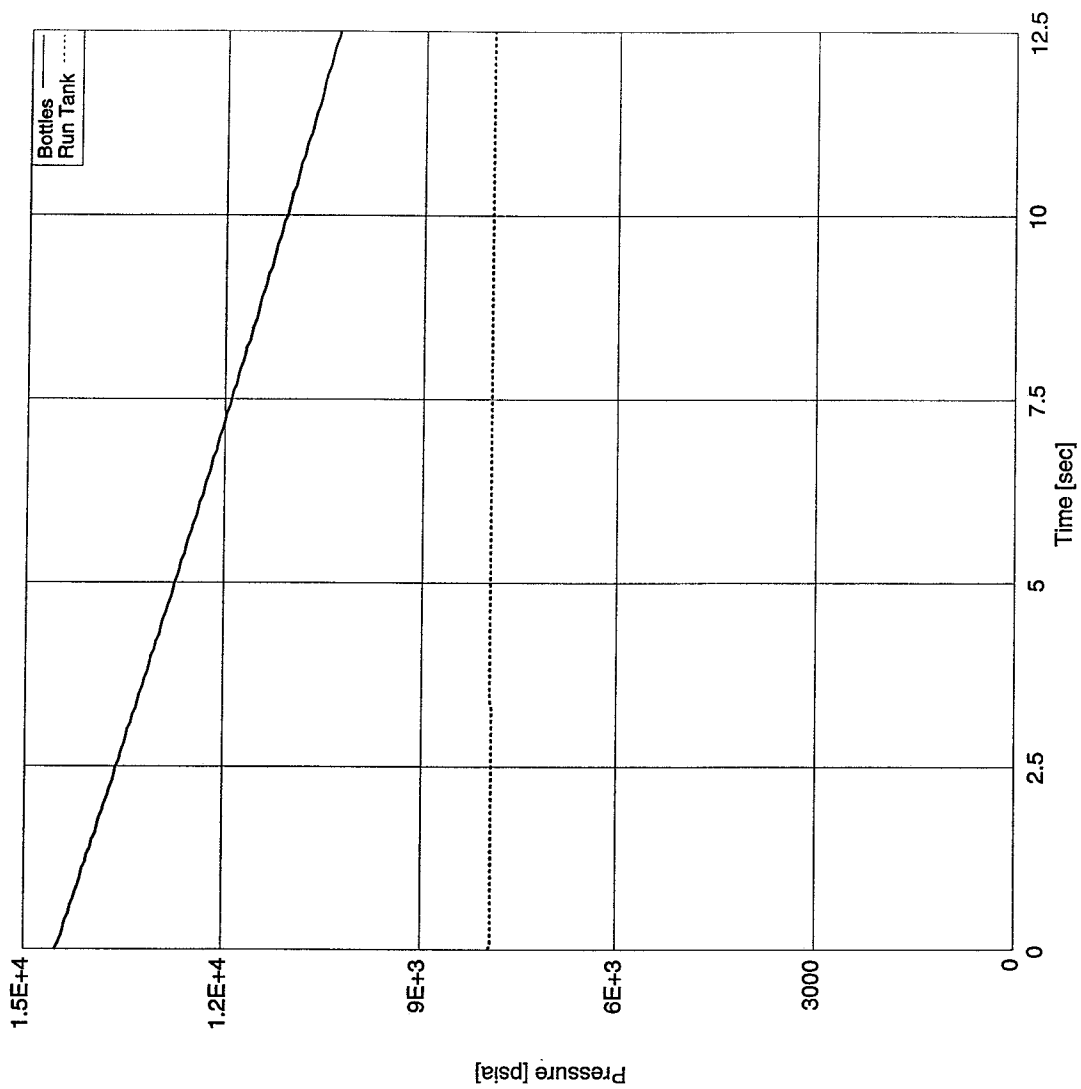


Fig. 7. Old High-Pressure O2 Model Pressure Histories.

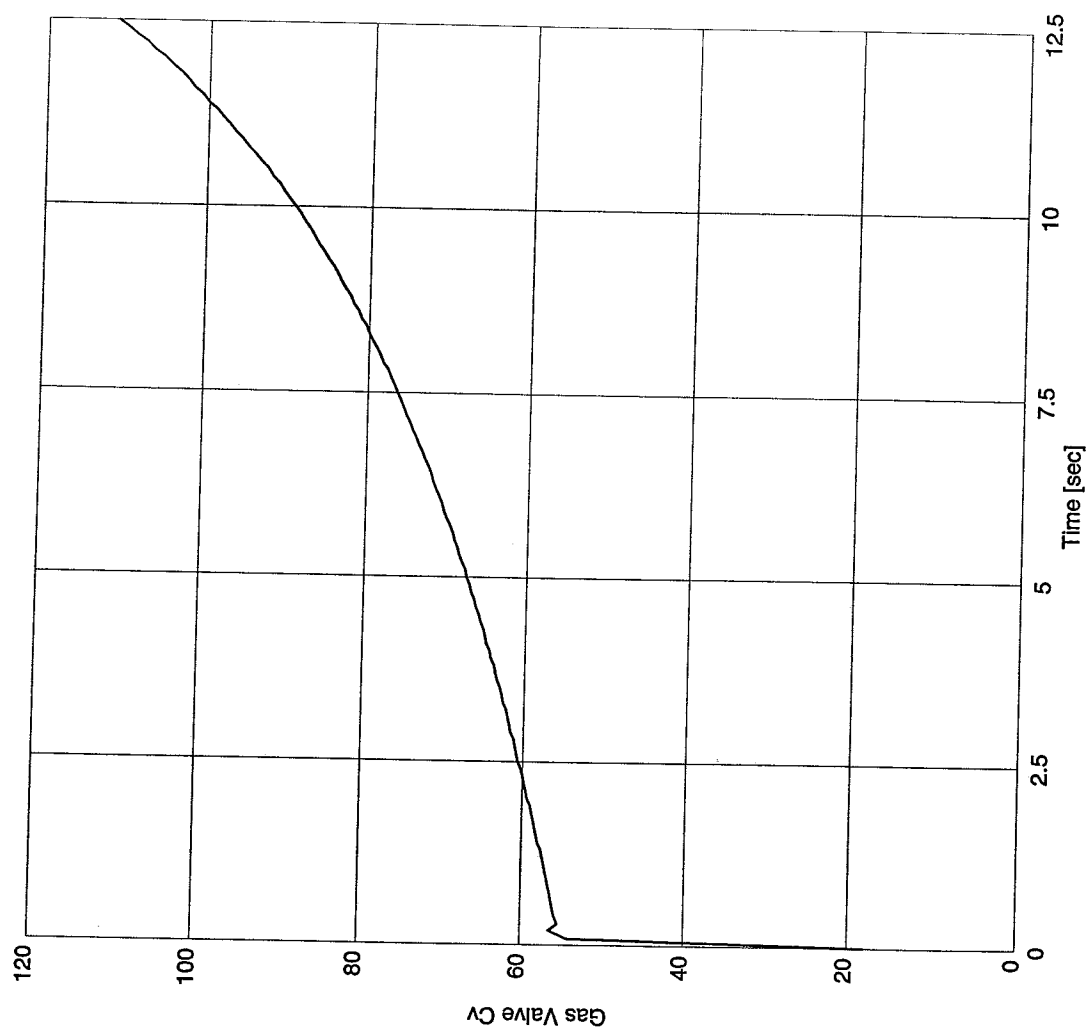


Fig. 8. Old High-Pressure O2 Model Gas Flow Control Valve Cv History.

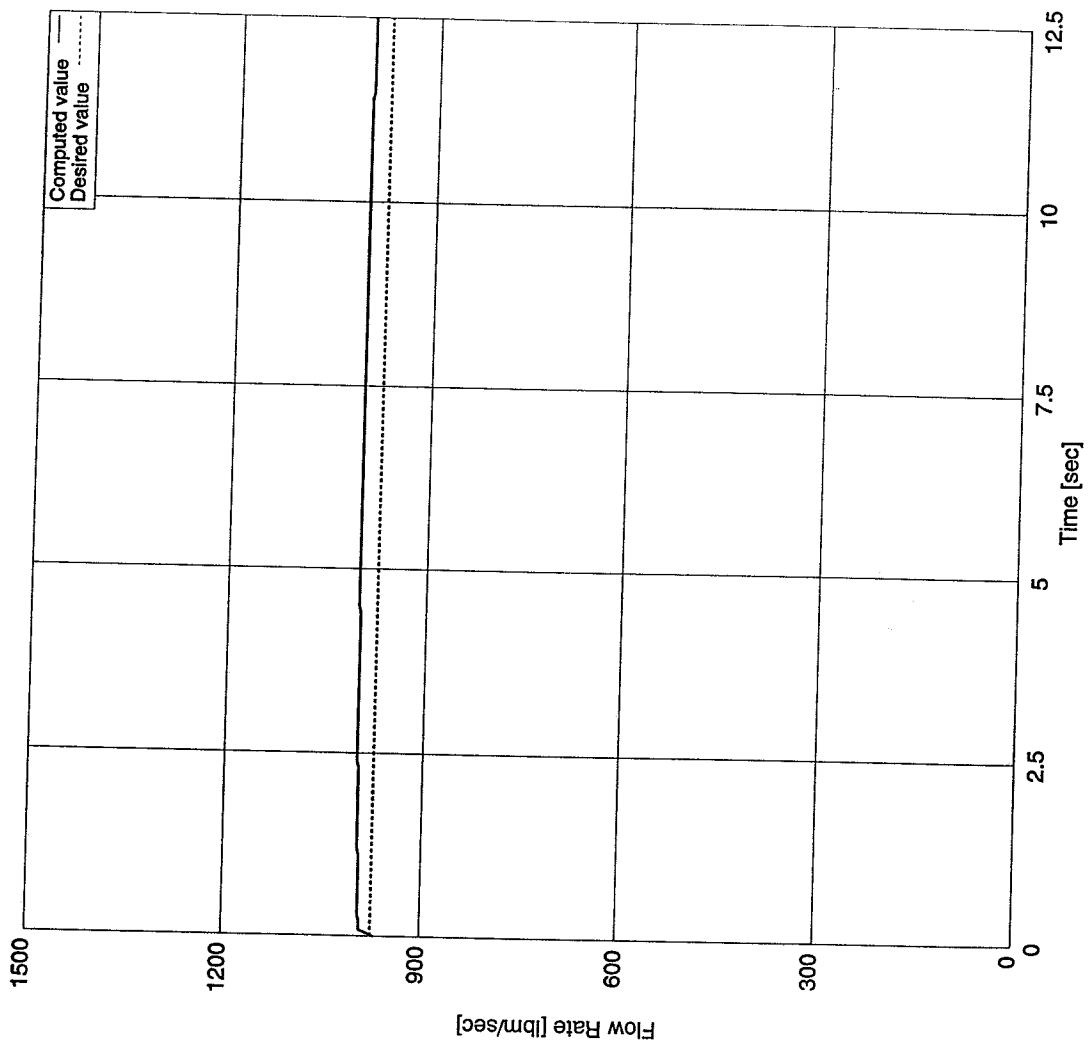


Fig. 9. Old High-Pressure O2 Model Liquid Flow Rate History.

NIST-12 Code Modifications

The 1995 version of the NIST-12 property routines, which was supplied to us by Mr. Steve Poulton of Lockheed SSC, would not handle mixed liquid-vapor states. Furthermore, it would not compute saturation properties accurately in some cases. The codes have been modified to allow mixed-phase states. More importantly, a typographical error in the NIST-12 FORTRAN program was discovered, and once corrected, the code now computes properties on or near the dome accurately and robustly.

Table 2 shows the FORTRAN code for the modified subroutine FPROP. This subroutine drives all of the property calculations and is the one, which is directly called by EASY/ROCETS. The modifications are highlighted in bold italic type. The mixed phase calculations are based around the concept of vapor quality

$$qual = \frac{\text{mass vapor}}{\text{total mass}}$$

Therefore, $qual = 1$ for a saturated vapor, and $qual = 0$ for a saturated liquid. For mixed states $0 < qual < 1.0$. Here the convention is established that for all nonsaturated states such as gases, compressed liquids, or supercritical states $qual$ will be set to -1 ($qual = -1.0$). In the mixed states all intensive properties (specific volume, $1/\rho$, internal energy, u , enthalpy, h , and entropy, s) are computed using the quality

$$\frac{1}{\rho} = qual \cdot \frac{1}{\rho_v} + (1 - qual) \cdot \frac{1}{\rho_L}$$

$$u = qual \cdot u_v + (1 - qual) \cdot u_L$$

$$h = qual \cdot h_v + (1 - qual) \cdot h_L$$

$$s = qual \cdot s_v + (1 - qual) \cdot s_L$$

This is not necessarily true for the mixed phase values of the other properties. Specific heats, c_p and c_v , and viscosity may not have meaning for mixed phases. However, they are naively computed using the formulas above. The mixed-phase values for d_i , c_p , c_v , v , μ , and so should not be used without further study.

When calls are made for saturated properties at a given pressure or temperature the user must supply the quality. When calls are made using intensive properties, such as pressure and internal energy or density and internal energy, the code tests for saturated conditions and computes quality.

Very small modifications to the FORTRAN functions FINDP and FINDT were also necessary and are shown in Tables 3 and 4.

Table 5 shows the FORTRAN code for the function VPN. This function computes the vapor pressure at a given temperature. The highlighted line, shown here in the corrected form, had a typographical error. The term $VP(2)$ was repeated twice as $VP(2)*VP(2)$. This caused significant inaccuracies for some materials, e.g. hydrogen. The uncorrected code would not compute saturation properties correctly and was not robust computing properties near the dome since the improper saturation properties were being used as starting guesses in many of the iterative functions, FINDD for example.

Table 6 shows before and after results for saturated hydrogen vapor at 50 psia along with values from the "Hydrogen Technological Survey Thermophysical Properties" tables (McCarty, 1975). From the table, we see that the uncorrected routines failed badly. First the saturation temperature is computed incorrectly. This error causes mass confusion with the other routines and the properties come out to be those for compressed liquid at 50.0 psia and 35.6 R. The computed properties after the correction agree very

closely with those given in the Tables. The energy values, u and h , are in slight error because the routines compute these saturated values at very small superheat to maintain the robustness of the iterative procedures imbedded in the code. The old “miprops” program does the same thing and gets very close to the same results as the after column in the table.

Figure 11 shows the input stream of the NIST-12 program for a mixed-phase state of N_2 at 50 psia and a quality of 0.5. Figure 12 shows the results for the same state when the inputs are density and internal energy. Notice that in this case a starting temperature guess is now required. If this option has trouble converging, try a different temperature guess.

The corrected code has proven to be very robust in that it usually converges. The one exception is near the critical point. Figure 13 shows a temperature-density-pressure map for parahydrogen (McCarty, 1975). The heart of the NIST database is a curve fit of this map in the form $p = f(t, \rho)$. This formula is used in an iterative mode in several places in the code. Inspections of the figure show that the isobars have an inflection point at the critical point. This causes the iterative algorithms to be unstable very near the critical point. Two fixes have been thought of. One is to box the critical point with four nearby points and to use interpolation when inside the box. The other is to use a bisection iterative scheme instead of the Newton’s method currently used in NIST-12.

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

C***** Subroutine *****
      subroutine fprop(ifl,iop,iu,p,t,qual,d,u,h,s,cp,cv,v,
& th,so,di)
      implicit real*8(a-h,o-z), integer*4(i-n)
C***** COMMON Structures *****
      common /crit/  em,      eok,      rm,      tc,      dc,
&                  x,       pc,       sig
      common /data/  g(32),   r,       gam,      vp(9),   dtp,
&                  pcc,     ptp,     tcc,     ttp,     tul,
&                  tll,     pul,     dcc
      common /con/   pcon,    tcon,    dcon,    ucon,    scon,
&                  vcon,    thcon,   socon
      common /satfl/ il,     ip,      isat,    dv,      dl
C***** End of COMMON Structures *****

      loop = 2
      if (ifl.eq.1) then
        call PH2()
      else if (ifl.eq.2) then
        call N2()
      else if (ifl.eq.3) then
        call O2()
      else if (ifl.eq.4) then
        call AR()
      else if (ifl.eq.5) then
        call NF3()
      else if (ifl.eq.6) then
        call CH4()
      else if (ifl.eq.7) then
        call C2H6()
      else if (ifl.eq.8) then
        call C2H4()
      else if (ifl.eq.9) then
        call C3H8()
      else if (ifl.eq.10) then
        call ISOB()
      else if (ifl.eq.11) then
        call NORB()
      else if (ifl.eq.12) then
        call D2()
      else if (ifl.eq.13) then
        call HE()
      else if (ifl.eq.14) then
        call CO2()
      else if (ifl.eq.15) then
        call CO()
      else if (ifl.eq.16) then
        call NH2()
      else if (ifl.eq.17) then
        call XE()
      else
        write(*,*) 'UNKNOWN FLUID NUMBER PLEASE REENTER'
        return
      endif
C*****
      if (iu .eq. 0) then
c      Engr. Units - Conversions
        pcon = 14.695949d0/0.101325d0
        tcon = 1.8d0
        dcon = em/16.0184637d0
        ucon = 1./(2.324445d0*em)
        scon = 1./(4.184001d0*em)
        vcon = 1.d-6*0.671875652d0

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

      thcon = 0.578176d0
      socon = 3.28084d0
    else
c      SI Units - Conversions
      pcon = 1.d0
      tcon = 1.d0
      dcon = em
      ucon = 1.d0/em*1.0d3
      scon = 1.d0/em*1.0d3
      vcon = 1.0d-3
      thcon = 1.d0
      socon = 1.d0
    endif
c      write(*,*) pcc*pcon, tcc*tcon
      pp = p/pcon
      tt = t/tcon
      dd = d/dcon
      uu = u/ucon
      hh = h/ucon
      ss = s/scon
      cpp = cp/scon
      cvv = cv/scon
      soo = so/socon
      vv = v/vcon
      thh = th/thcon
c+++++
      if (iop .eq. 0) then
        call satur( pp, tt, ip )
        if (il .lt. 0) then
          plus = 0.025d-02*pp
          dl = findd(pp+plus,tt)
          dv = findd(pp-plus,tt)
          call energy( pp, tt, dv, uv, hv, sv )
          call spheat( tt, dv, cppv, cvvv, soov )
          div = dielec( pp, tt, dv )
          call visthe( dv, tt, vv, thhv )
          call energy( pp, tt, dl, ul, hl, sl )
          call spheat( tt, dl, cppl, cvvl, sool )
          dil = dielec( pp, tt, dl )
          call visthe( dl, tt, vl, thhl )
          dd=qual*(1.0d0/dv)+(1.0d0-qual)*(1.0d0/dl)
          dd=1.0d0/dd
          uu=qual*uv+(1.0d0-qual)*ul
          hh=qual*hv+(1.0d0-qual)*hl
          ss=qual*sv+(1.0d0-qual)*sl
          di=qual*div+(1.0d0-qual)*dil
          cpp=qual*cppv+(1.0d0-qual)*cppl
          cvv=qual*cvvv+(1.0d0-qual)*cvvl
          vv=qual*vvv+(1.0d0-qual)*vvl
          thh=qual*thhv+(1.0d0-qual)*thhl
          soo=qual*soov+(1.0d0-qual)*sool
        else
          if (il .eq. 0) then
            plus = 0.25d-01*pp
            dd = findd( pp+plus, tt )
            qual = 0.0d0
          else
            plus = 0.25d-02*pp
            dd = findd( pp-plus, tt )
            qual = 1.0d0
          endif
          call energy( pp, tt, dd, uu, hh, ss )
          call spheat( tt, dd, cpp, cvv, soo )

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
    end if
  else if (iop .eq. 1) then
    call limits( pp, tt )
    dd = findd( pp, tt )
    call energy( pp, tt, dd, uu, hh, ss )
    call spheat( tt, dd, cpp, cvv, soo )
    di = dielec( pp, tt, dd )
    call visthe( dd, tt, vv, thh )
    qual = -1.0
  else if (iop .eq. 2) then
    il = -1
    isat = 0
    tt = findt( pp, dd )
    call limits( pp, tt )
    if (isat.eq.1) then
      call energy( pp, tt, dv, uv, hv, sv )
      call spheat( tt, dv, cppv, cvvv, soov )
      div = dielec( pp, tt, dv )
      call visthe( dv, tt, vvv, thhv )
      call energy( pp, tt, dl, ul, hl, sl )
      call spheat( tt, dl, cppl, cvvl, sool )
      dil = dielec( pp, tt, dl )
      call visthe( dl, tt, vvl, thhl )
      qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
      uu=qual*uv+(1.0d0-qual)*ul
      hh=qual*hv+(1.0d0-qual)*hl
      ss=qual*sv+(1.0d0-qual)*sl
      di=qual*div+(1.0d0-qual)*dil
      cpp=qual*cppv+(1.0d0-qual)*cppl
      cvv=qual*cvvv+(1.0d0-qual)*cvvl
      vv=qual*vvv+(1.0d0-qual)*vvl
      thh=qual*thhv+(1.0d0-qual)*thhl
      soo=qual*soov+(1.0d0-qual)*sool
    else
      call energy( pp, tt, dd, uu, hh, ss )
      call spheat( tt, dd, cpp, cvv, soo )
      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
      qual = -1.0d0
    end if
  else if (iop .eq. 3) then
    il = -1
    isat = 0
    pp = findp( tt, dd )
    call limits( pp, tt )
    if (isat.eq.1) then
      call energy( pp, tt, dv, uv, hv, sv )
      call spheat( tt, dv, cppv, cvvv, soov )
      div = dielec( pp, tt, dv )
      call visthe( dv, tt, vvv, thhv )
      call energy( pp, tt, dl, ul, hl, sl )
      call spheat( tt, dl, cppl, cvvl, sool )
      dil = dielec( pp, tt, dl )
      call visthe( dl, tt, vvl, thhl )
      qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
      uu=qual*uv+(1.0d0-qual)*ul
      hh=qual*hv+(1.0d0-qual)*hl
      ss=qual*sv+(1.0d0-qual)*sl
      di=qual*div+(1.0d0-qual)*dil
      cpp=qual*cppv+(1.0d0-qual)*cppl
      cvv=qual*cvvv+(1.0d0-qual)*cvvl
    end if
  end if

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

      vv=qual*vvv+(1.0d0-qual)*vv1
      thh=qual*thhv+(1.0d0-qual)*thh1
      soo=qual*soov+(1.0d0-qual)*sool
    else
      call energy( pp, tt, dd, uu, hh, ss )
      call spheat( tt, dd, cpp, cvv, soo )
      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
      qual = -1.0d0
    end if
  else if (iop .eq. 4) then
    isat = 0
    if((pp.lt.pcc).and.(pp.gt.ptp)) then
      ip = 1
      call satur(pp,tsat,ip)
      dv = findd(pp-1.0d-04,tsat)
      dl = findd(pp+0.25d-02*pp,tsat)
      call energy( pp, tsat, dv, uv, hv, sv )
      call energy( pp, tsat, dl, ul, hl, sl )
      if((ss.lt.sv).and.(ss.gt.sl)) isat = 1
    end if
    if (isat.eq.1) then
      tt = tsat
      call spheat( tt, dv, cppv, cvvv, soov )
      div = dielec( pp, tt, dv )
      call visthe( dv, tt, vvv, thhv )
      call spheat( tt, dl, cppl, cvvl, sool )
      dil = dielec( pp, tt, dl )
      call visthe( dl, tt, vvl, thhl )
      qual = (ss - sl)/(sv - sl)
      uu=qual*uv+(1.0d0-qual)*ul
      hh=qual*hv+(1.0d0-qual)*hl
      dd=qual*(1.0d0/dv)+(1.0d0-qual)*(1.0d0/dl)
      dd=1.0d0/dd
      di=qual*div+(1.0d0-qual)*dil
      cpp=qual*cppv+(1.0d0-qual)*cppl
      cvv=qual*cvvv+(1.0d0-qual)*cvvl
      vv=qual*vvv+(1.0d0-qual)*vv1
      thh=qual*thhv+(1.0d0-qual)*thh1
      soo=qual*soov+(1.0d0-qual)*sool
    else
      si = ss
      isig = 0
      do while (isig .lt. 4)
        dd = findd( pp, tt )
        call energy( pp, tt, dd, uu, hh, ss )
        err = ss-si
        call nuite( tt, err, np, isig, loop )
      end do
      if (isig .eq. 4) then
        write(*,*) 'FPROP - Option ',iop,' Failed to Converge'
      endif
      dd = findd( pp, tt )
      call limits( pp, tt )
      call energy( pp, tt, dd, uu, hh, ss )
      call spheat( tt, dd, cpp, cvv, soo )
      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
    end if
  else if (iop .eq. 5) then
    ui = uu
    isig = 0
    il = -1

```


Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

isat = 0
do while (isig .lt. 4)
  pp = findp( tt, dd )
  if (isat.eq.1) then
    call energy(pp,tt,dv,uv,hv,sv)
    call energy(pp,tt,dl,ul,hl,sl)
    qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
    uu=qual*uv+(1.0d0-qual)*ul
  else
    call energy( pp, tt, dd, uu, hh, ss )
  end if
  err = uu-ui
  call nuite( tt, err, np, isig, loop )
end do
if (isig .eq. 4) then
  write(*,*) 'FPROP - Option ',iop,' Failed to Converge'
endif
pp = findp( tt, dd )
call limits( pp, tt )
if (isat.eq.1) then
  call energy( pp, tt, dv, uv, hv, sv )
  call spheat( tt, dv, cppv, cvvv, soov )
  div = dielec( pp, tt, dv )
  call visthe( dv, tt, vvv, thhv )
  call energy( pp, tt, dl, ul, hl, sl )
  call spheat( tt, dl, cppl, cvvl, sool )
  dil = dielec( pp, tt, dl )
  call visthe( dl, tt, vvl, thhl )
  qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
  uu=qual*uv+(1.0d0-qual)*ul
  hh=qual*hv+(1.0d0-qual)*hl
  ss=qual*sv+(1.0d0-qual)*sl
  di=qual*div+(1.0d0-qual)*dil
  cpp=qual*cppv+(1.0d0-qual)*cppl
  cvv=qual*cvvv+(1.0d0-qual)*cvvl
  vv=qual*vvv+(1.0d0-qual)*vvl
  thh=qual*thhv+(1.0d0-qual)*thhl
  soo=qual*soov+(1.0d0-qual)*sool
else
  call energy( pp, tt, dd, uu, hh, ss )
  call spheat( tt, dd, cpp, cvv, soo )
  di = dielec( pp, tt, dd )
  call visthe( dd, tt, vv, thh )
  qual = -1.0d0
end if
else if (iop .eq. 6) then
  isat = 0
  if((pp.lt.pcc).and.(pp.gt.ptp)) then
    ip = 1
    call satur(pp,tsat,ip)
    dv = findd(pp-1.0d-04,tsat)
    dl = findd(pp+0.25d-02*pp,tsat)
    call energy( pp, tsat, dv, uv, hv, sv )
    call energy( pp, tsat, dl, ul, hl, sl )
    if((hh.lt.hv).and.(hh.gt.hl)) isat = 1
  end if
  if (isat.eq.1) then
    tt = tsat
    call spheat( tt, dv, cppv, cvvv, soov )
    div = dielec( pp, tt, dv )
    call visthe( dv, tt, vvv, thhv )
    call spheat( tt, dl, cppl, cvvl, sool )
    dil = dielec( pp, tt, dl )
  end if
end if

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

      call visthe( dl, tt, vvl, thhl )
      qual = (hh - hl)/(hv - hl)
      uu=qual*uv+(1.0d0-qual)*ul
      ss=qual*sv+(1.0d0-qual)*sl
      dd=qual*(1.0d0/dv)+(1.0d0-qual)*(1.0d0/dl)
      dd=1.0d0/dd
      di=qual*div+(1.0d0-qual)*dil
      cpp=qual*cppv+(1.0d0-qual)*cppl
      cvv=qual*cvvv+(1.0d0-qual)*cvvl
      vv=qual*vvv+(1.0d0-qual)*vv1
      thh=qual*thhv+(1.0d0-qual)*thhl
      soo=qual*soov+(1.0d0-qual)*sool
    else
      hi = hh
      isig = 0
      do while (isig .lt. 4)
        dd = findd( pp, tt )
        call energy( pp, tt, dd, uu, hh, ss )
        err = hh-hi
        call nutter( tt, err, np, isig, loop )
      end do
      if (isig .eq. 4) then
        write(*,*) 'FPROP - Option ',iop,' Failed to Converge'
      endif
      dd = findd( pp, tt )
      call limits( pp, tt )
      call energy( pp, tt, dd, uu, hh, ss )
      call spheat( tt, dd, cpp, cvv, soo )
      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
      qual = -1.0d00
    end if
  else if (iop .eq. 7) then
    si = ss
    isig = 0
    il = -1
    isat = 0
    do while (isig .lt. 4)
      pp = findp( tt, dd )

      if (isat.eq.1) then
        call energy(pp,tt,dv,uv,hv,sv)
        call energy(pp,tt,dl,ul,hl,sl)
        qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
        ss=qual*sv+(1.0d0-qual)*sl
      else
        call energy( pp, tt, dd, uu, hh, ss )
      end if
      err = ss-si
      call nutter( tt, err, np, isig, loop )

    end do
    if (isig .eq. 4) then
      write(*,*) 'FPROP - Option ',iop,' Failed to Converge'
    endif
    isat = 0
    pp = findp( tt, dd )
    call limits( pp, tt )
    if (isat.eq.1) then
      call energy( pp, tt, dv, uv, hv, sv )
      call spheat( tt, dv, cppv, cvvv, soov )
      div = dielec( pp, tt, dv )
      call visthe( dv, tt, vvv, thhv )
    end if
  end if

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

call energy( pp, tt, dl, ul, hl, sl )
call spheat( tt, dl, cppl, cvvl, sool )
dil = dielec( pp, tt, dl )
call visthe( dl, tt, vvl, thhl )
qual = (1.0d0/dd - 1.0d0/dl)/(1.0d0/dv - 1.0d0/dl)
uu=qual*uv+(1.0d0-qual)*ul
hh=qual*hv+(1.0d0-qual)*hl
ss=qual*sv+(1.0d0-qual)*sl
di=qual*div+(1.0d0-qual)*dil
cpp=qual*cppv+(1.0d0-qual)*cppl
cvv=qual*cvvv+(1.0d0-qual)*cvvl
vv=qual*vvv+(1.0d0-qual)*vvl
thh=qual*thhv+(1.0d0-qual)*thhl
soo=qual*soov+(1.0d0-qual)*sool
else
call energy( pp, tt, dd, uu, hh, ss )
call spheat( tt, dd, cpp, cvv, soo )
di = dielec( pp, tt, dd )
call visthe( dd, tt, vv, thh )
qual = -1.0d0
end if
else if (iop .eq. 8) then
isat = 0
if((pp.lt.pcc).and.(pp.gt.ptp)) then
ip = 1
call satur(pp,tsat,ip)
dv = findd(pp-1.0d-04,tsat)
dl = findd(pp+0.25d-02*pp,tsat)
call energy( pp, tsat, dv, uv, hv, sv )
call energy( pp, tsat, dl, ul, hl, sl )
if((uu.lt.uv).and.(uu.gt.ul)) isat = 1
end if
if (isat.eq.1) then
tt = tsat
call spheat( tt, dv, cppv, cvvv, soov )
div = dielec( pp, tt, dv )
call visthe( dv, tt, vvv, thhv )
call spheat( tt, dl, cppl, cvvl, sool )
dil = dielec( pp, tt, dl )
call visthe( dl, tt, vvl, thhl )
qual = (uu - ul)/(uv - ul)
dd=qual*(1.0d0/dv)+(1.0d0-qual)*(1.0d0/dl)
dd=1.0d0/dd
hh=qual*hv+(1.0d0-qual)*hl
ss=qual*sv+(1.0d0-qual)*sl
di=qual*div+(1.0d0-qual)*dil
cpp=qual*cppv+(1.0d0-qual)*cppl
cvv=qual*cvvv+(1.0d0-qual)*cvvl
vv=qual*vvv+(1.0d0-qual)*vvl
thh=qual*thhv+(1.0d0-qual)*thhl
soo=qual*soov+(1.0d0-qual)*sool
else
ui = uu
isig = 0
do while (isig .lt. 4)
dd = findd( pp, tt )
call energy( pp, tt, dd, uu, hh, ss )
err = uu-ui
call nutter( tt, err, np, isig, loop )
end do
if (isig .eq. 4) then
write(*,*) 'FPROP - Option ',iop,' Failed to Converge'
endif

```

Table 2. FORTRAN Code for Modified NIST_12 FPROP Subroutine

```

      dd = findd( pp, tt )
      call limits( pp, tt )
      call energy( pp, tt, dd, uu, hh, ss )
      call spheat( tt, dd, cpp, cvv, soo )
      di = dielec( pp, tt, dd )
      call visthe( dd, tt, vv, thh )
      qual = -1.d00
    end if
  else
    pp = 0.d0
    tt = 0.d0
    dd = 0.d0
    uu = 0.d0
    hh = 0.d0
    ss = 0.d0
    cpp = 0.d0
    cvv = 0.d0
    vv = 0.d0
    thh = 0.d0
    soo = 0.d0
    di = 0.d0
  endif
c+++++*
  p = pp*pcon
  t = tt*tcon
  d = dd*dcon
  u = uu*ucon
  h = hh*ucon
  s = ss*scon
  cp = cpp*scon
  cv = cvv*scon
  so = soo*socon
  v = vv*vcon
  th = thh*thcon

  return
end
c***** End Subroutine *****

```

Table 3. FORTRAN Code for Modified NIST_12 FINDP Function

```

c***** Function *****
"
  real*8 function findp(t, d)
"
    implicit real*8(a-h,o-z), integer*4(i-n)
c+++++ COMMON Structures +++++
    common /data/  g(32),  r,      gam,    vp(9),  dtp,    pcc,
&                ptp,    tcc,    ttp,    tul,    tll,    pul,
&                dcc
    common /satfl/ il,    ip,    isat,  dv,    d1
c+++++ End of COMMON Structures +++++
    ieq = 1
    isat = 0
    d1 = d
    if ( (t .lt. tcc) .and. (il .lt. 0) ) then
        p = vpn( t )
        pp = p - 1.d-4
        dv = findd( pp, t )
        pp = p + 0.25d-02*pp
        d1 = findd( pp, t )
        if (d .gt. dv .and. d .lt. d1) then
            d1 = dv
            isat = 1
c
            write(*,*)
c            write(*,*) 'findp - The state point specified corresponds'
c            write(*,*) '          to a Density in the 2-phase region'
c            write(*,*) '          SATURATED VAPOP IS ASSUMED'
c            write(*,*)
c        endif
    endif
    findp = props( d1, t, ieq )
    return
end
c***** end Function *****

```

Table 4. FORTRAN Code for Modified NIST_12 FINDT Function

```

C***** Function *****
"
  real*8 function findt(p, d)
"
  implicit real*8(a-h,o-z), integer*4(i-n)
"
C***** COMMON Structures *****
  common /data/ g(32), r, gam, vp(9), dtp, pcc,
&               ptp, tcc, ttp, tul, tll, pul,
&               dcc
  common /con/ pcon, tcon, dcon, ucon, scon, vcon,
&             thcon, socon
  common /satfl/ il, ip, isat, dv, dl
C***** end of COMMON Structures *****
C*****
C* Returns Temperature(K), from the 32-term MBWR eqn of state. *
C* input is Pressure(mPa) and Density(mol/l). *
C*****
  ieq = 1
  loop = 1
  isat = 0
  if (p .gt. pcc) then
    tt = tcc
  else
    tsat = tcc
    isig = 0
    do while (isig .lt. 4)
      pp = vpn( tsat )
      err = p - pp
      call nuiter ( tsat, err, np, isig, loop )
    end do
    if (isig .eq. 4) then
      write(*,*)
      write(*,*) 'findt - Vapor pressure equ. Failed to converge'
      write(*,*) '          PRESS = ',p*pcon,' TSAT = ',tsat*tcon
      write(*,*) '          ERROR = ',(p-pp)*pcon
      write(*,*)
    endif
    pp = p - 1.d-4
    dv = findd( pp, tsat )
    pp = p + 0.25d-02*p
    dl = findd( pp, tsat )
    if ( (d .gt. dv) .and. (d .lt. dl) ) then
      isat = 1
      write(*,*)
      write(*,*) 'findt - The state point specified corresponds'
      write(*,*) '          to a Density in the 2-phase region'
      write(*,*)
      findt = tsat
      return
    endif
    tt = tsat
  endif
  isig = 0
  do while (isig .lt. 4)
    pp = props( d, tt, ieq )
    err = p - pp
    call nuiter ( tt, err, np, isig, loop )
  end do
  if (isig .eq. 4) then
    write(*,*)
    write(*,*) 'findt - Equation of state Failed to converge'
    write(*,*) '          PRESS = ',p*pcon,' TEMP = ',tt*tcon
  endif

```

Table 4. FORTRAN Code for Modified NIST_12 FINDT Function

```

        write(*,*) '          ERROR = ', (p-pp)*pcon
        write(*,*)
    endif
    findt = tt
    return
end
c***** end Function *****

```

Table 5. FORTRAN Code for Modified NIST_12 VPN FUNCTION

```

c***** Function *****
      real*8 function vpn(y)
      implicit real*8(a-h,o-z), integer*4(i-n)
c+++++ COMMON Structures ++++++
      common /data/ g(32), r,      gam,      vp(9), dtp,      pcc,
&      ptp,      tcc,      ttp,      tul,      tll,      pul,
&      dcc
c+++++ end of COMMON Structures ++++++
      xz = (1.d0-vp(7)/y)/(1.d0-vp(7)/vp(8))
c
      vpn = vp(1) * xz +
&      vp(2) * (xz**2.d0) +
&      vp(3) * (xz**3.d0) +
&      vp(4) * (xz**4.d0) +
&      vp(5) * xz * ( dabs(1.d0-xz)**vp(6) )
      vpn = vp(9) * dexp(vpn)
c
      return
      end
c***** end Function *****

```

Table 6. Comparison of Corrected and Uncorrected Computations for Saturated Hydrogen Vapor at 50 psia.

	Tables	Before	After
Pressure [psia]	50.0	50.0	50.0
Temperature [R]	45.4	35.6	45.4
Density [lbm/ft ³]	0.262	4.47	0.261
Internal Energy [BTU/lbm]	52.3	-113.2	52.8
Enthalpy [BTU/lbm]	87.6	-111.1	88.2
Entropy [BTU/lbm-R]	6.29	1.84	6.30


```

*****
*
*               NIST Standard Reference Database #12
*       NIST Thermophysical Properties of Pure Fluids Database
*
*
*   This Program Provides the Thermodynamic Properties of 17 Fluids
*
*   Para Hydrogen,   Nitrogen,   Oxygen,
*   Argon,           Nitrogen Trifluoride,   Methane,
*   Ethane,          Ethylene,   Propane,
*   Iso Butane,      Normal Butane,   Deuterium,
*   Helium,          Carbon Dioxide,   Carbon Monoxide,
*   Normal Hydrogen, Xenon
*
*
*       Original Development by R.D. McCarty and V. Arp
*       Based on correlations described by NBS Technical Note 1097
*
*               Revised - May 15, 1995
*
*****
Is the Input coming from a file? (y/n)
      Default - (n) :
n
      Select A Fluid By Entering The Corresponding Number
*****
*   1=Para Hydrogen      2=Nitrogen      3=Oxygen      *
*   4=Argon              5=Nitrogen Trifluoride  6=Methane      *
*   7=Ethane            8=Ethylene      9=Propane      *
*  10=Iso Butane        11=Normal Butane  12=Deuterium   *
*  13=Helium            14=Carbon Dioxide  15=Carbon Monoxide *
*  16=Normal Hydrogen   17=Xenon      18=Stop       *
*****
      Default (18) - Stop :
2
For Engineering Units Enter "0", for Metric Units   Enter "1"
      Default 0 - Engineering Units :
0
      Enter The Input Option Desired
*****
*   0=Saturation Properties      1=Input Pressure & Temperature      *
*   2=Input Pressure & Density    3=Input Temperature & Density      *
*   4=Input Pressure & Entropy    5=Input Density & Internal Energy   *
*   6=Input Pressure & Enthalpy   7=Input Density & Entropy      *
*   8=Input Pressure & Internal Energy
*****
      Default 1 - Pressure & Temperature :
0
For Liquid Enter "0", For Vapor Enter "1"
For Mixture Enter "-1"
      Default 0 - Liquid :
-1
Input Temperature Enter "0", Input Pressure "1"
      Default 0 - Temperature :
1
*****
To Select Another Fluid Enter Zero (0) for the Input

```

Figure 11. Output Listing for Modified NIST_12 Property Routines with Saturated Fluid Capability-- Pressure and Quality as Input.

```

Enter a Pressure in (psia)
(Decimal Point Required)
50.
Enter the Quality
(Decimal Point Required)
.5

*****
*   Press  =      50.0 (psia)                                *
*   Temp   =     161.11 (deg R)                               *
*   Temp   =    -298.56 (deg R)                               *
*   Qual    =      0.500                                       *
*   Dens    =      1.768 (lbm/ft3)          Z =  0.45818      *
*   IntEng  =     -7.60 (Btu/lbm)                               *
*   Enthal  =     -2.36 (Btu/lbm)                               *
*   Entrop  =      0.992 (Btu/lbm-R)                            *
*   Cv      =      0.209 (Btu/lbm-R)                            *
*   Cp      =      0.404 (Btu/lbm-R)          k =  1.93094    *
*   Sound   =     1532. (ft/sec)                                *
*   Visc    =     3.57968 xE-5 (lbm/ft-sec)      =  0.05328 (c-Poise) *
*   Th Con  =     0.03461 (Btu/ft-hr-R)                               *
*   Diel    =      1.20256                                       *
*****

```

Figure 11. Output Listing for Modified NIST_12 Property Routines with Saturated Fluid Capability-- Pressure and Quality as Input.

```

Enter The Input Option Desired
*****
* 0=Saturation Properties      1=Input Pressure & Temperature *
* 2=Input Pressure & Density   3=Input Temperature & Density *
* 4=Input Pressure & Entropy   5=Input Density & Internal Energy *
* 6=Input Pressure & Enthalpy  7=Input Density & Entropy *
* 8=Input Pressure & Internal Energy *
*****
Default 1 - Pressure & Temperature :
5
*****

To Select Another Fluid Enter Zero (0) for the Input
Enter Density (lbm/ft3) and Internal Ergy (Btu/lbm) Temp Guess (deg R)
(Decimal Points and Comma Required)
1.768,-7.6,170.

*****
* Press = 50.0 (psia) *
* Temp = 161.11 (deg R) *
* Temp = -298.56 (deg R) *
* Qual = 0.500 *
* Dens = 1.768 (lbm/ft3) Z = 0.45809 *
* IntEng = -7.60 (Btu/lbm) *
* Enthal = -2.37 (Btu/lbm) *
* Entrop = 0.992 (Btu/lbm-R) *
* Cv = 0.209 (Btu/lbm-R) *
* Cp = 0.404 (Btu/lbm-R) k = 1.93089 *
* Sound = 1532. (ft/sec) *
* Visc = 3.57973 xE-5 (lbm/ft-sec) = 0.05328 (c-Poise) *
* Th Con = 0.03461 (Btu/ft-hr-R) *
* Diel = 1.20255 *
*****

```

Figure 12. Output Listing for Modified NIST_12 Property Routines with Saturated Fluid Capability-- Density and Internal Energy as Input.

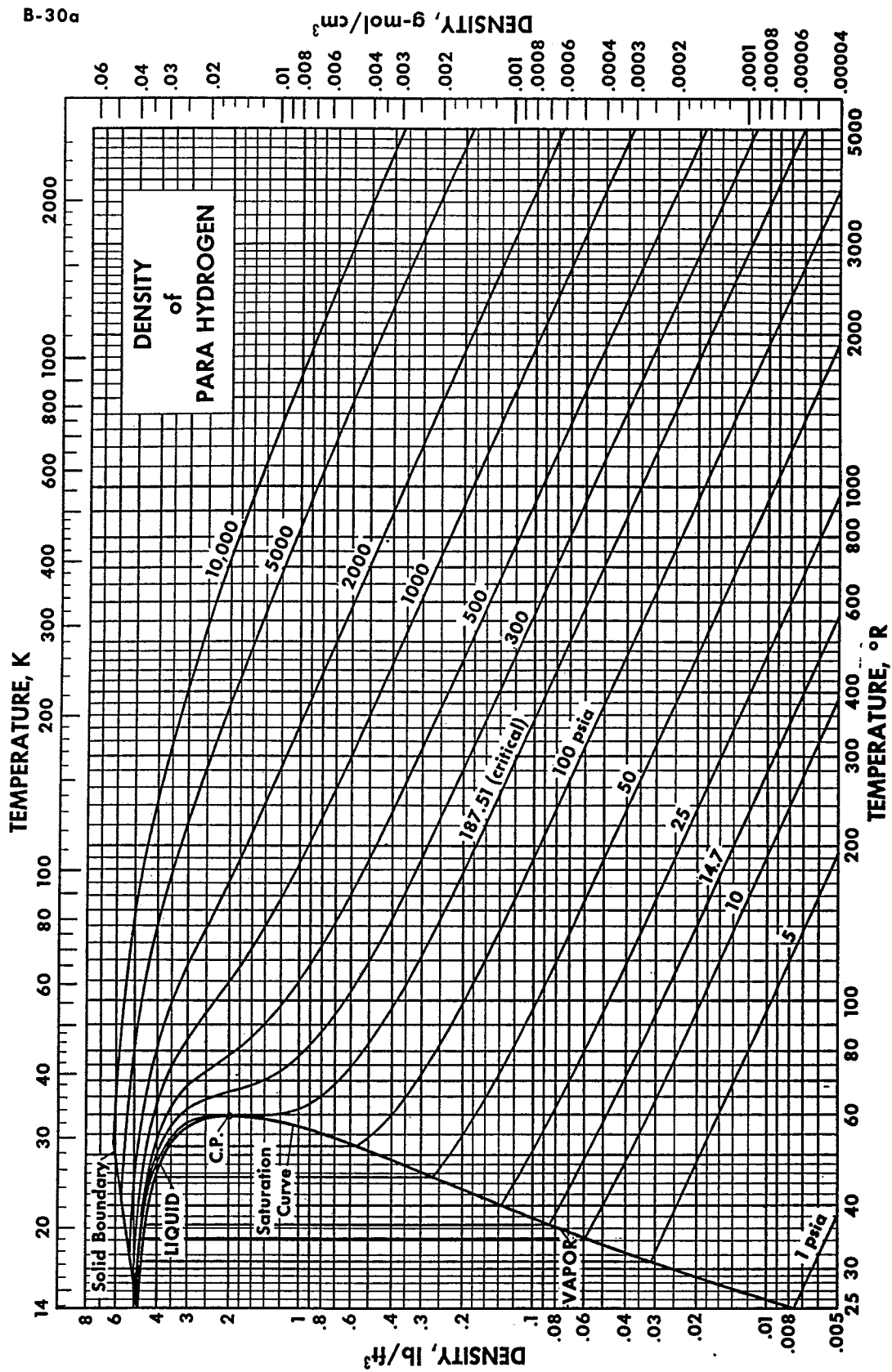


Fig. 13. Parahydrogen Temperature-Density-Pressure Map

“nr” Library—Revised For NIST-12

Using the NIST-12 routines to make all property calculations allows opportunities for initial condition input that were not available with the ROCETS table lookup routines. Also, in the original ROCETS routines each property was a separate table lookup call. The NIST-12 routines are organized to compute all of the properties with each call to FPROP. Therefore, it would be inefficient to paste the NIST-12 codes into the old ROCETS paradigm as was done in the revised “er” library discussed above. This prompted us to completely revise the modules to be used with the NIST-12 routines. This new library is abbreviated “nr” in the EASY5x library list.

The organization of the new library can be fully understood by reviewing four modules. The first is the bottle module. Figure 14 shows the input window for the bottle module. Looking at the “Inputs” column, it is seen that the initial state is input by specifying the temperature, pressure, and quality (TVG, PVG, and VQG) instead of the pressure and enthalpy in the “er” library. Pressure and temperature were selected as the more natural properties to define the initial state. When the quality is set equal to -1 , the default, both pressure and temperature are used to compute the state properties. When quality is set between 0 and 1, saturation properties are computed at the specified pressure, and the temperature input is ignored. The fluid (H2, N2, O2) is chosen with the map variable. Care should be taken with the map variable. In line with the option numbers in NIST-12 (1=H2, 2=N2, 3=O2) the map numbers have changed from the “er” library. Table 7 shows the EASY5 macro FORTRAN code for the nr-library bottle.

Figure 15 shows the input window for the inlet module. Again the Input column contains pressure, temperature, and quality to set the initial state properties. Table 8

contains the FORTRAN macro code. Figure 16 shows the input window for pipe module PIPE99, PA. It should be noted that the Input column no longer has a place for the map variable since the viscosity is now fed in through the inlet port from the neighboring volume-type element. Table 9 contains the FORTRAN code.

Figure 18 shows a test model for some pipe and valve elements along with the flow rate obtained with both the old “er” library and the new “nr” library. The results are virtually identical.

Figure 19 shows another test model that tests the bottle and other elements. Figures 20 and 21 show time histories of the bottle pressure, the volume pressure, and the two flow rates from the “nr” library calculations. Figures 22 and 23 show the same thing for the “er” library. The plots are for all purposes identical.

Figure 24 shows a test model that includes a runtank. Figures 25 and 26 show plots of the tank pressure, gas flow rate, and liquid flow rate for computations using both libraries. The plots are almost identical. The numbers on the gas flow rate are very slightly different, but the change in gas flow rate with time is the same.

Figure 27 is a test model that considers saturated fluids with mixed-phase states. This case is similar to the case considered in Figure 3. Figures 28 and 29 show pressure and temperature histories and the two flow rates. These figures can be compared with Figures 4 and 5. The results are seen to be similar. The quality in Figure 28 is seen to approach 1 and to switch to -1 as the last liquid in the volume boils off.

[illegible]

Table 7. Macro Library for nr Library Bottle Module

```

C
C      1 = H2
C      2 = N2
C      3 = O2
C      13 = He
C
stop sort
      IF(ICCALC .EQ. 1) THEN
C      Use initial values of pressure and temperature to set IC's
        P B0-- = PVGB0--
        TTTB0-- = TVGB0--
        VQ B0-- = VQGB0--
        IF (VQ B0-- .LT. 0.0D00) THEN
          CALL FPROP(NINT(MAPB0--), 1, IUNIT,
1          P B0--, TTTB0--, VQ B0--, d, UT B0--,
2          HT B0--, S, CP B0--, CV, V, th,
3          SO, di)
        ELSE IF ((VQ B0-- .GE. 0.0D00) .AND. (VQ B0-- .LE. 1.0D00)) THEN
          IP = 1
          IL = -1
          CALL FPROP(NINT(MAPB0--), 0, IUNIT,
1          P B0--, TTTB0--, VQ B0--, d, UT B0--,
2          HT B0--, S, CP B0--, CV, V, th,
3          SO, di)
        ELSE
          WRITE(*,*)
          WRITE(*,*) 'IMPROPER VALUE FOR VQ B0--'
          STOP
        END IF
        CALL EZSETV(RHOB0--, d/CFOOT**3)
      ELSE
C      Compute properties based on u and rho
        d = RHOB0-- * CFOOT**3
        CALL FPROP(NINT(MAPB0--), 5, IUNIT,
1          P B0--, TTTB0--, VQ B0--, d, UT B0--,
2          HT B0--, S, CP B0--, CV, V, th,
3          SO, di)
        ENDIF
        P B0-- = P B0--
        VQ B0-- = VQ B0--
        PI1B0-- = P B0--
        HT B0-- = HT B0--
        HI1B0-- = HT B0--
        TTTB0-- = TTTB0--
        TI1B0-- = TTTB0--
C
        CP B0-- = CP B0--
C
        GM B0-- = CP B0--/CV
        GI1B0-- = GM B0--
C
        SI1B0-- = S
C
        RM B0-- = V/cvis
        MI1B0-- = RM B0--
C
        RG B0-- = (CP B0-- - CV) * RJ
        RG1B0-- = RG B0--
C
        RH1B0-- = RHOB0--
resume sort

```


Table 8. Macro Library for nr Library Inlet Module

```

C  MACRO source for "IN" - easyrockets exit module
C  Get the map index
C    1 = para hydrogen , H2
C    2 = nitrogen      , N2
C    3 = oxygen        , O2
C   13 = Helium        , He
stop sort
C
C  compute properties as functions of P and T
C
  VQ IN-- = VQGIN--
  IF(VQ IN--.LT.0.0D00) THEN
    NIST1 = JIDNNT(MAPIN--)
    NIST2 = 1
    NIST3 = IUNIT
    CALL FPROP(NIST1, NIST2, NIST3, PININ--, TTTIN--, VQ IN--,
1      d, UININ--, HININ--, S, CP IN--, CV, RM IN--, th,
2      SO, di)
  ELSE IF((VQ IN--.GE.0.0D00).AND.(VQ IN--.LE.1.0D00)) THEN
    IP = 1
    IL = -1
    NIST1 = JIDNNT(MAPIN--)
    NIST2 = 0
    NIST3 = IUNIT
    CALL FPROP(NIST1, NIST2, NIST3, PININ--, TTTIN--, VQ IN--,
1      d, UININ--, HININ--, S, CP IN--, CV, RM IN--, th,
2      SO, di)

  ELSE
    WRITE(*,*)
    WRITE(*,*) 'IMPROPER VALUE FOR VQ IN--'
    STOP
  END IF
  PI1IN-- = PININ--
  VQ IN-- = VQ IN--
  HI1IN-- = HININ--
  RH1IN-- = d / CFOOT ** 3
  GI1IN-- = CP IN--/CV
  RG1IN-- = (CP IN-- - CV)*RJ
  MI1IN-- = RM IN--/CVIS
  TI1IN-- = TTTIN--
  W1 IN-- = W IN--
resume sort

```

Table 9. Macro Library for nr Library Pipe99 Module

```

C      SUBROUTINE PIPE99 ( IPRPL , IUPDAT , MODN , NOD1 ,
C      $                  NOD2 , DIA , ERUF , RKF ,
C      $                  RLEN , PTIN , PTOUT , RHOIN ,
C      $                  RHOOUT , RMUIN , RMUOUT , W )
C      implicit real*8(a-h,o-z)
C*****
C %BEGIN CLASS PIPE99
C
C      SUBPROGRAM PIPE99          UNCLASSIFIED
C
C
C      BOB TAYLOR
C      STENNIS SPACE CENTER/MISS. STATE UNIV
C
C %END CLASS PIPE99
C*****
C %BEGIN PURPOSE PIPE99
C
C      CALCULATE INCOMPRESSIBLE FLUID FLOW THROUGH A PIPE
C      WITH A LOSS--FLOW COEFFICIENT INTERNALLY COMPUTED.
C      (ADOPTED FROM PIPE01)
C
C %END PURPOSE PIPE99
C*****
C %BEGIN HISTORY PIPE99
C
C      WRITTEN                      06/09/93 BOB TAYLOR
C
C %END HISTORY PIPE99
C*****
C %BEGIN SCHEMATIC PIPE99
C
C
C      NOD1      ---> W      NOD2
C
C %END SCHEMATIC PIPE99
C*****
stop sort
      RMIPA-- = MO2PA--
      RMOPA-- = MI1PA--
      CALL PIPE99 ( nint(PRPPA--), 0 , 'PA--' , 'NOD1' ,
$                  'NOD2' , DIAPA-- , ERFPA-- , RKFPA-- ,
$                  RL PA-- , PI1PA-- , PO2PA-- , RH1PA-- ,
$                  RH2PA-- , RMIPA-- , RMOPA-- , W PA-- )

      W1 PA-- = W PA--
      W2 PA-- = W PA--
resume sort
stop sort
      HI2PA-- = HI1PA--
      HO1PA-- = HO2PA--
      GI2PA-- = GI1PA--
      GO1PA-- = GO2PA--
      TI2PA-- = TI1PA--
      TO1PA-- = TO2PA--
resume sort

```

Table 9. Macro Library for nr Library Pipe99 Module

```

C      SUBROUTINE RTANK ( IPRPL , IUPDAT , MODN , NOD1 ,
C      $                  NOD2 , NOD3 , NOD4 , NOD5 ,
C      $                  NOD6 , VTOT , TWALLI , HTIN ,
C      $                  HTOUT , WIN , WOUT , PV ,
C      $                  TV , HTV , RKV , TL ,
C      $                  HTL , RKL , BETAV , RNUV ,
C      $                  ALPV , BETAL , RNUL , ALPL ,
C      $                  RHOV , VVAP , UTV , UTL ,
C      $                  DRVDT , DVVDT , DUVDT , DULDT ,
C      $                  RHOL)
C*****
C %BEGIN CLASS RTANK
C
C      SUBPROGRAM RTANK              UNCLASSIFIED
C
C
C
C      BOB TAYLOR
C      STENNIS SPACE CENTER/MISSISSIPPI STATE UNIV.
C
C %END CLASS RTANK
C*****
C %BEGIN PURPOSE RTANK
C
C      THIS MODULE MODELS THE DYNAMICS OF A RUN TANK
C      PRESSURIZATION WITH VAPOR ULLAGE. INDEPENDANT
C      MASS AND ENERGY BALANCES FOR THE VAPOR AND LIQUID
C      ARE USED TO DETERMINE DENSITY AND INTERNAL ENERGY
C      DERIVATIVES.
C
C %END PURPOSE RTANK
C*****
C %BEGIN HISTORY RTANK
C
C      WRITTEN                      05/29/96
C
C %END HISTORY RTANK
C*****
C %BEGIN SCHEMATIC RTANK
C
C
C
C      NOD1 WIN
C      NOD2 HTIN
C      NOD3 VAPOR
C      PROP3
C
C      VAPOR
C
C      LIQUID
C
C      NOD4 WOUT
C      NOD5 HTOUT
C      NOD6 LIQ
C      PROP5
C
C %END SCHEMATIC RTANK
C*****
C
stop sort
C
C      VAPOR SECTION
C
C      The fluid map indexes are
C
C      1 = H2

```

Table 9. Macro Library for nr Library Pipe99 Module

```

C      2 = N2
C      3 = O2
C      13 = He
C
      IF (ICCALC.EQ.1) THEN

        CALL FPROP(NINT(MPVRK--), 1, IUNIT, PVGRK--, TVGRK--, VQVRK--,
1         D,UTV, HTVRK--, S, CPVRK--, CV, V, TH, SO, DI)
        PV RK-- = PVGRK--
        TV RK-- = TVGRK--
        CALL EZSETV(RHVRK--, D/CFOOT**3)
        CALL EZSETV(UTVRK--,UTV)

      ELSE

C CONVERT UNITS ON RHO
      d = RHVRK-- * CFOOT**3
C
      CALL FPROP(NINT(MPVRK--), 5, IUNIT, PV RK--, TV RK--, VQVRK--,
1      D, UTVRK--, HTVRK--, S, CPVRK--, CV, V, TH, SO, DI)
      ENDIF
C
      VQVRK-- = VQVRK--
      RMVRK-- = V/CVIS
      RNVRK-- = RMVRK-- / RHVRK--
      RKVRK-- = TH / CTHK
      ALVRK-- = RKVRK-- / RHVRK-- / CPVRK--
      GMVRK-- = CPVRK-- / CV
      RGVK-- = (CPVRK-- - CV)*RJ
C
      TBVRK-- = TV RK-- + 0.001*TV RK--
      CALL FPROP(NINT(MPVRK--), 1, IUNIT, PV RK--, TV RK--, VQV1, D1,
1      U1, H1, S1, CP1, CV1, V1, TH1, SO1, DI1)
      CALL FPROP(NINT(MPVRK--), 1, IUNIT, PV RK--, TBVRK--, VQV2, D2,
1      U1, H1, S1, CP1, CV1, V1, TH1, SO1, DI1)
      BTVRK-- = (D2 - D1)/(TBVRK-- - TV RK--)/(-D1)

resume sort
stop sort
C
C LIQUID SECTION
C
C The fluid map indexes are
C
C      1 = H2
C      2 = N2
C      3 = O2
C      13 = He
C
      IF (ICCALC.EQ.1) THEN

        CALL FPROP(NINT(MPLRK--), 1, IUNIT, PVGRK--, TLGRK--, VQLRK--,
1         D, UTL, HTLRK--, S, CPLRK--, CV, V, TH, SO, DI)
        PL RK-- = PVGRK--
        TL RK-- = TLGRK--
        CALL EZSETV(UTLRK--,UTL)
        CALL EZSETV(VVPRK--,VGRK--)
      ELSE
        PL RK-- = PV RK--
        CALL FPROP(NINT(MPLRK--), 8, IUNIT, PL RK--, TL RK--, VQLRK--,
1         D, UTLRK--, HTLRK--, S, CPLRK--, CV, V, TH, SO, DI)
      ENDIF

```

Table 9. Macro Library for nr Library Pipe99 Module

```

RHLRK-- = D/CFOOT**3

VQLRK-- = VQLRK--
RMLRK-- = V/CVIS
RNLRK-- = RMLRK-- / RHLRK--
RKLrk-- = TH / CTHK
ALLRK-- = RKLrk-- / RHLRK-- / CPLRK--
GMLRK-- = CPLRK-- / CV
RGLRK-- = (CPLRK-- - CV)*RJ

TBLRK-- = TL RK-- - 0.001*TL RK--
CALL FPROP(NINT(MPLRK--), 1, IUNIT, PL RK--, TL RK--, VQL1, D1,
1      U1, H1, S1, CP1, CV1, V1, TH1, SO1, DI1)
CALL FPROP(NINT(MPLRK--), 1, IUNIT, PL RK--, TBLRK--, VQL2, D2,
1      U1, H1, S1, CP1, CV1, V1, TH1, SO1, DI1)
BTLRK-- = (D2 - D1)/(TBLRK-- - TL RK--)/(-D1)

resume sort
stop sort

C
  call RTANK ( NINT(PRPRK--), 0, 'RK--', 'NOD1', 'NOD2',
1      'NOD3', 'NOD4', 'NOD5', 'NOD6', VOLRK-- ,
1      TWIRK-- , HI1RK-- , HO2RK-- , W1 RK-- , W2 RK-- ,
1      PV RK-- , TV RK-- , HTVRK-- , RKVRK-- , TL RK-- ,
1      HTLRK-- , RKLrk-- , BTVRK-- , RNVRK-- , ALVRK-- ,
1      BTLRK-- , RNLrk-- , ALLRK-- , RHVRK-- , VVPRK-- ,
1      UTVRK-- , UTLRK-- , DRVRK-- , DVVRK-- , DUVRK-- ,
1      DULRK-- , RHLRK-- )

C
derivative of, RHVRK-- = DRVRK--
derivative of, VVPRK-- = DVVRK--
derivative of, UTVRK-- = DUVRK--
derivative of, UTLRK-- = DULRK--
C
resume sort
stop sort

RG1RK-- = RGVrk--
RG2RK-- = RGLRK--

PO1RK-- = PV RK--
PI2RK-- = PL RK--

RH1RK-- = RHVRK--
RH2RK-- = RHLRK--

HO1RK-- = HTVRK--
HI2RK-- = HTLRK--

TO1RK-- = TV RK--
TI2RK-- = TL RK--

GO1RK-- = GMVRK--
GI2RK-- = GMLRK--

MI1RK-- = RMVRK--
MI2RK-- = RMLRK--
RK1RK-- = RKVRK--
RK2RK-- = RKLrk--

resume sort

```


Component Data Table												
Options												
Component PA (or) PIPES from ROGETS (Pumping)												
Inputs				Output Status					Output Value			
Name	P	Value	Name	P	C Value	Frozen	Error	L	Name	P		
DIA	1	1.99999							W	1		
ERF	1	1.99999							W	2		
RKF	1	1.99999							HI	2		
RL	1	1.99999							HO	1		
PRP	1	0							GI	2		
PI	1	1.99999							GO	1		
PO	2	1.99999							TI	2		
RH	1	1.99999							TO	1		
RH	2	1.99999										
HI	1	1.99999										

Fig. 16. Input Window for nr-Library PIPE99 Module.

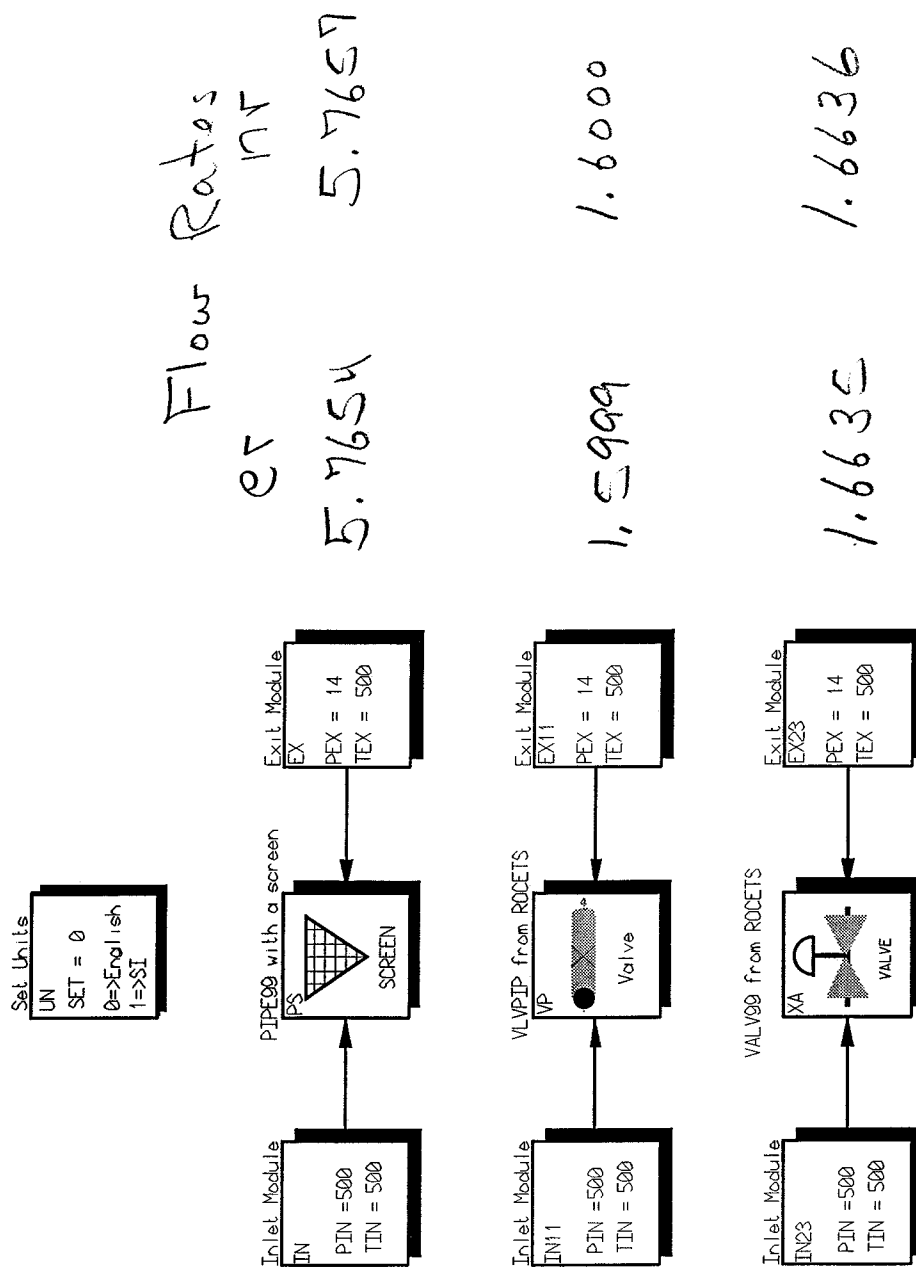


Fig. 18. Test Model for Valves and Pipes—nr Library

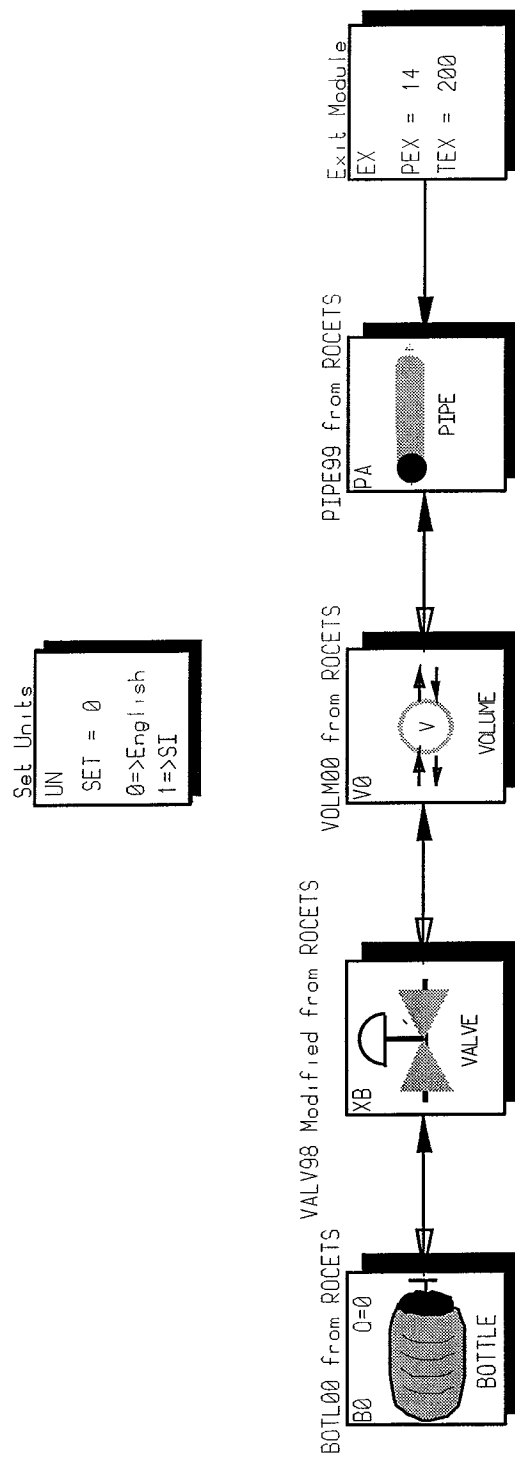


Fig. 19. Test Model for Bottle and Volume—nr Library

Bottle Test for nr Library

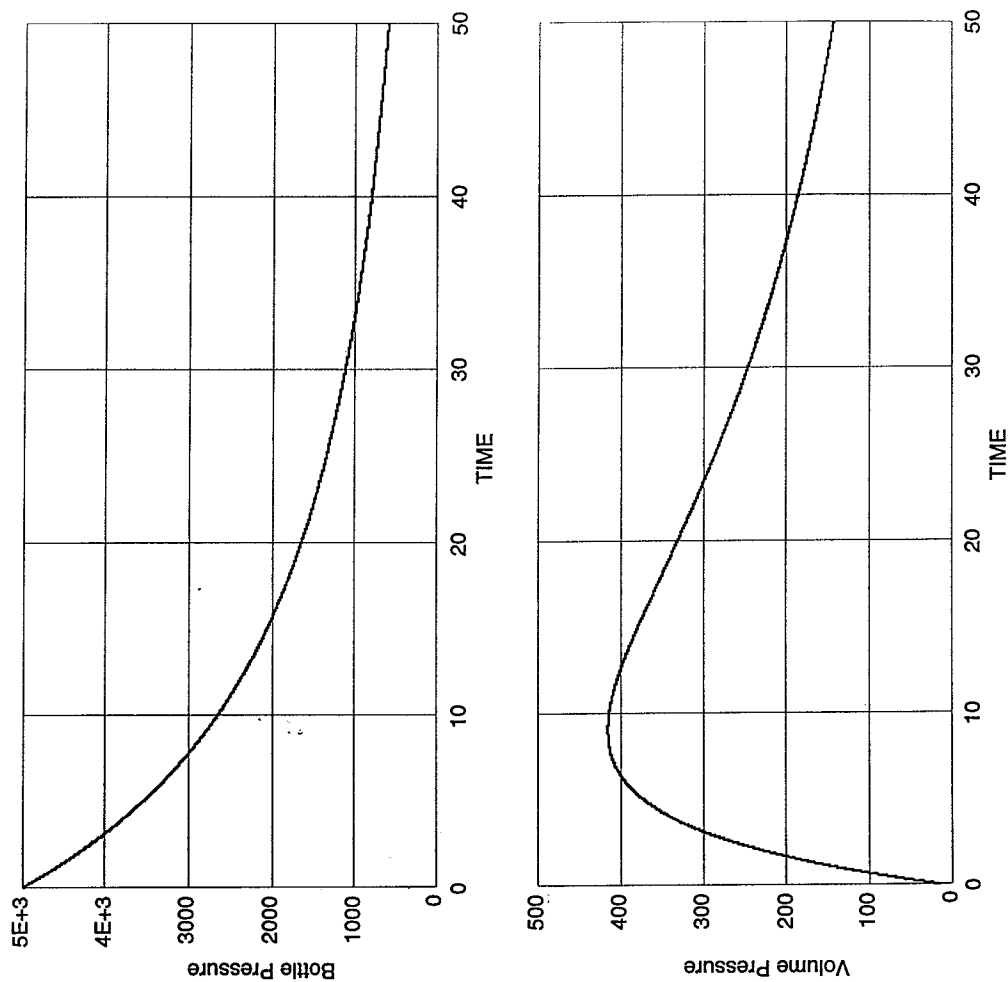


Fig. 20. Results of the Bottle Test Using the nr Library.

Bottle Test for nr Library

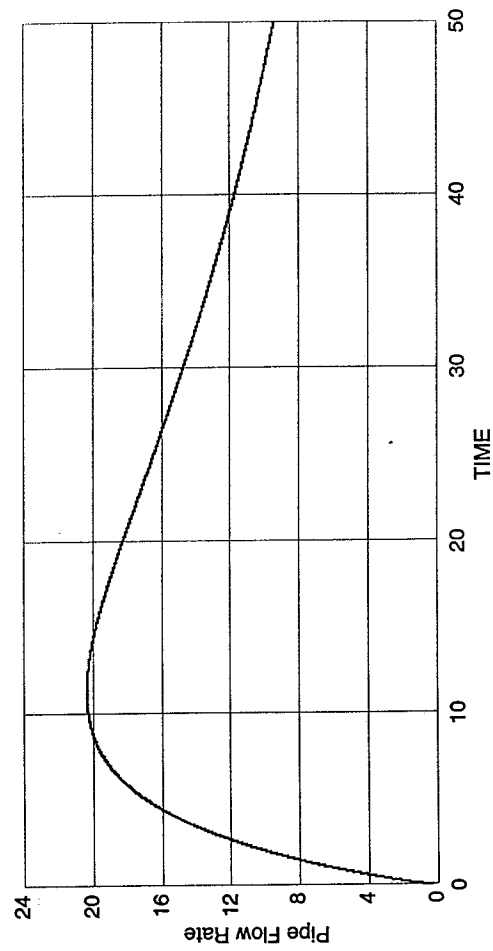
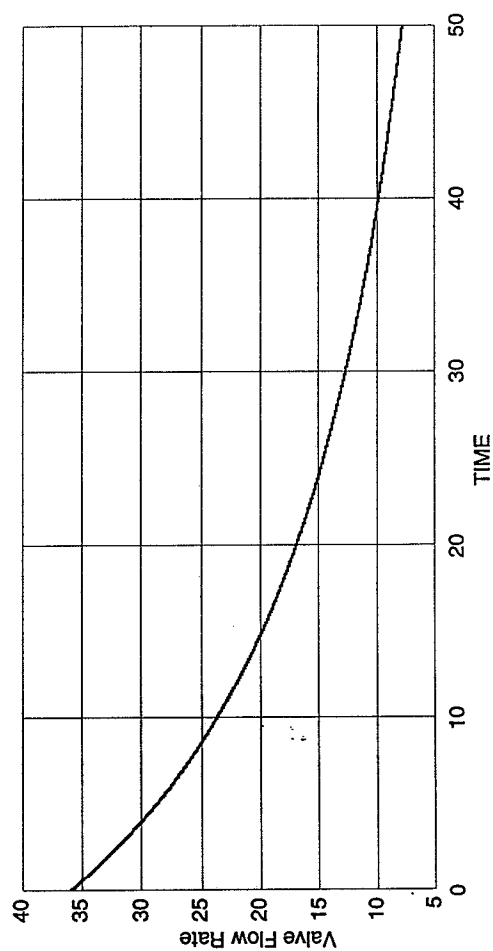


Fig. 21.. Results of the Bottle Test Using the nr Library.

Bottle Test for er Library

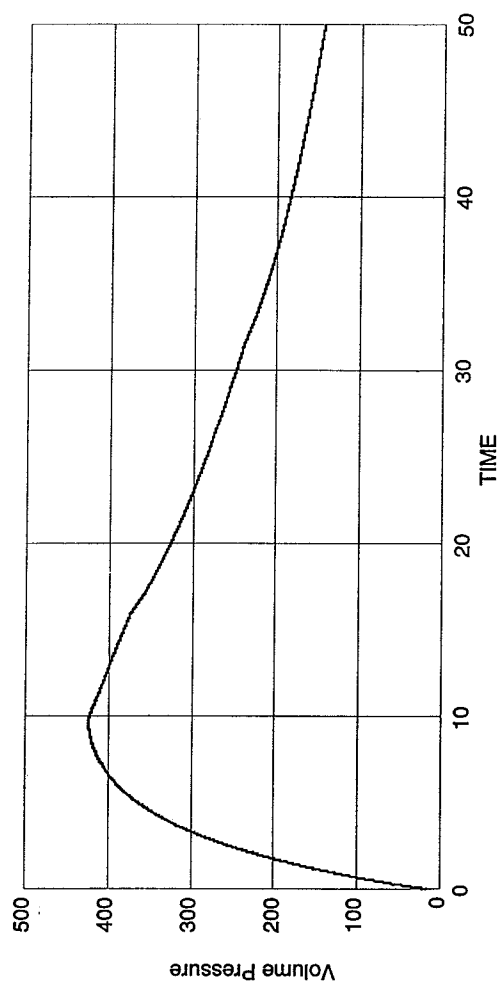
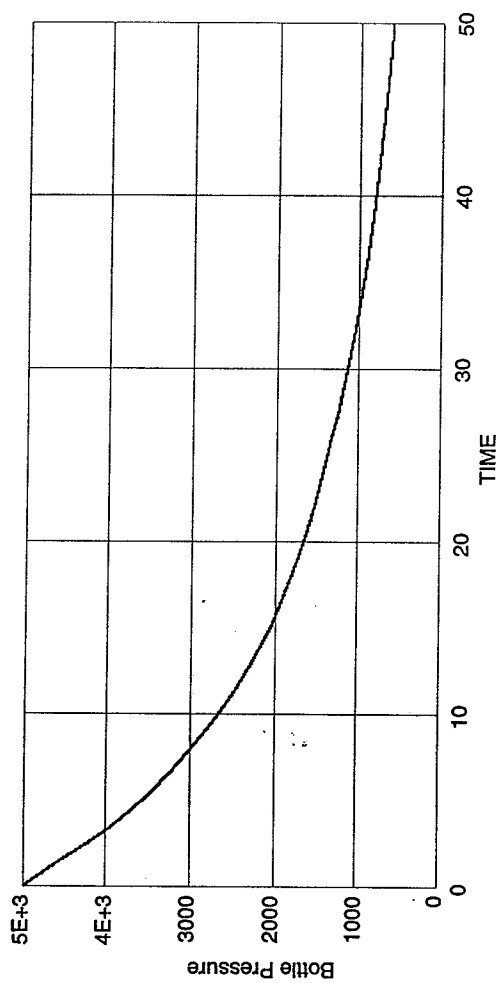


Fig. 22. Results of the Bottle Test Using the er Library.

Bottle Test for er Library

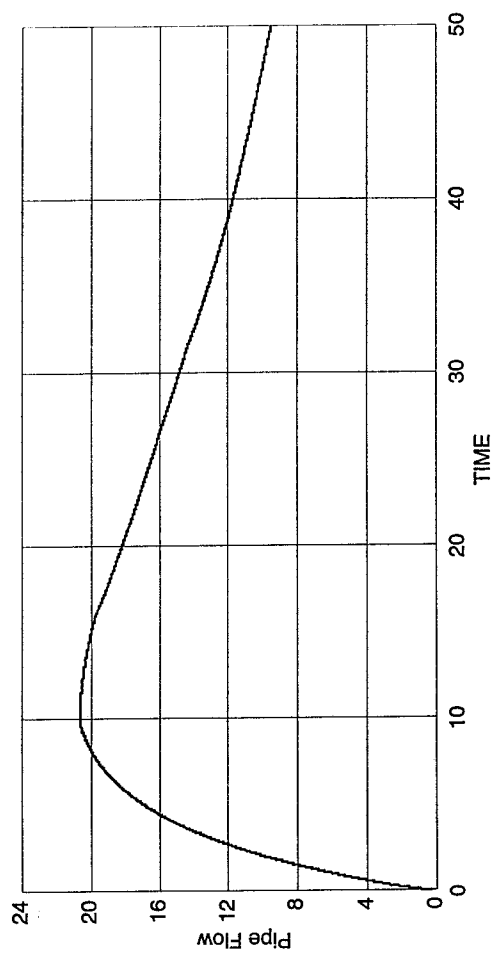
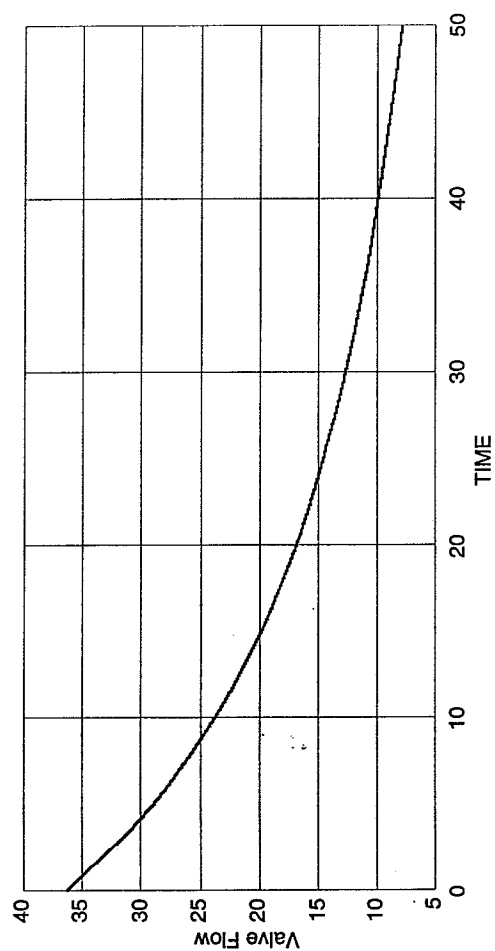


Fig. 23. Results of the Bottle Tests Using the er Library.

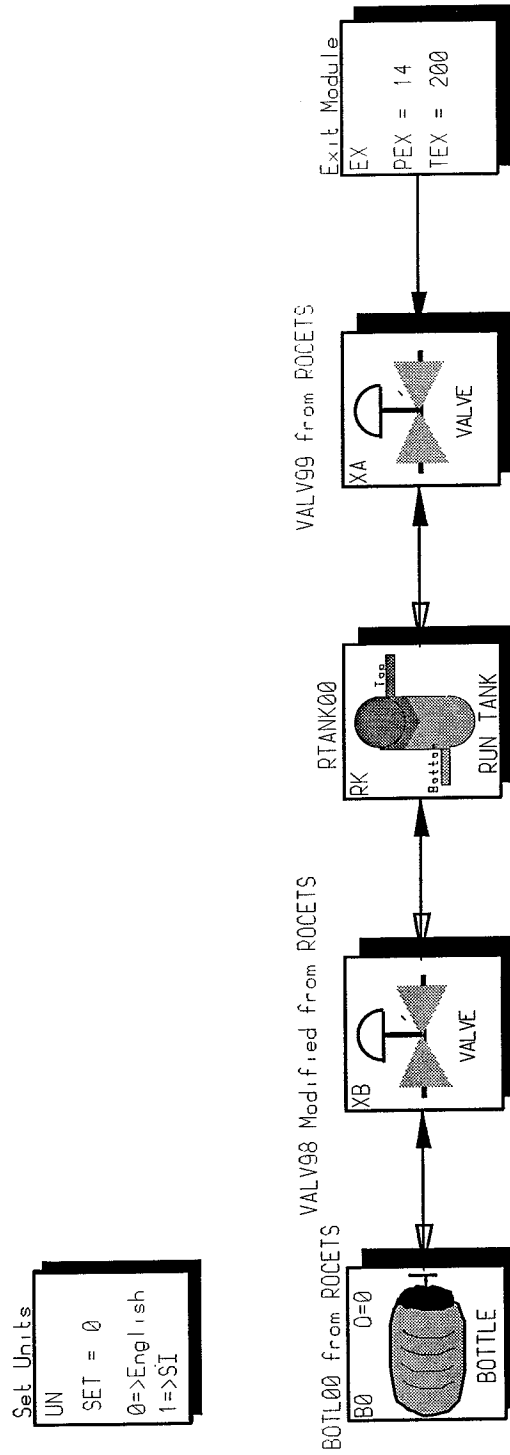


Fig. 24. Test Model Using a Run tank Element—nr Library.

Runtank Test for nr Library

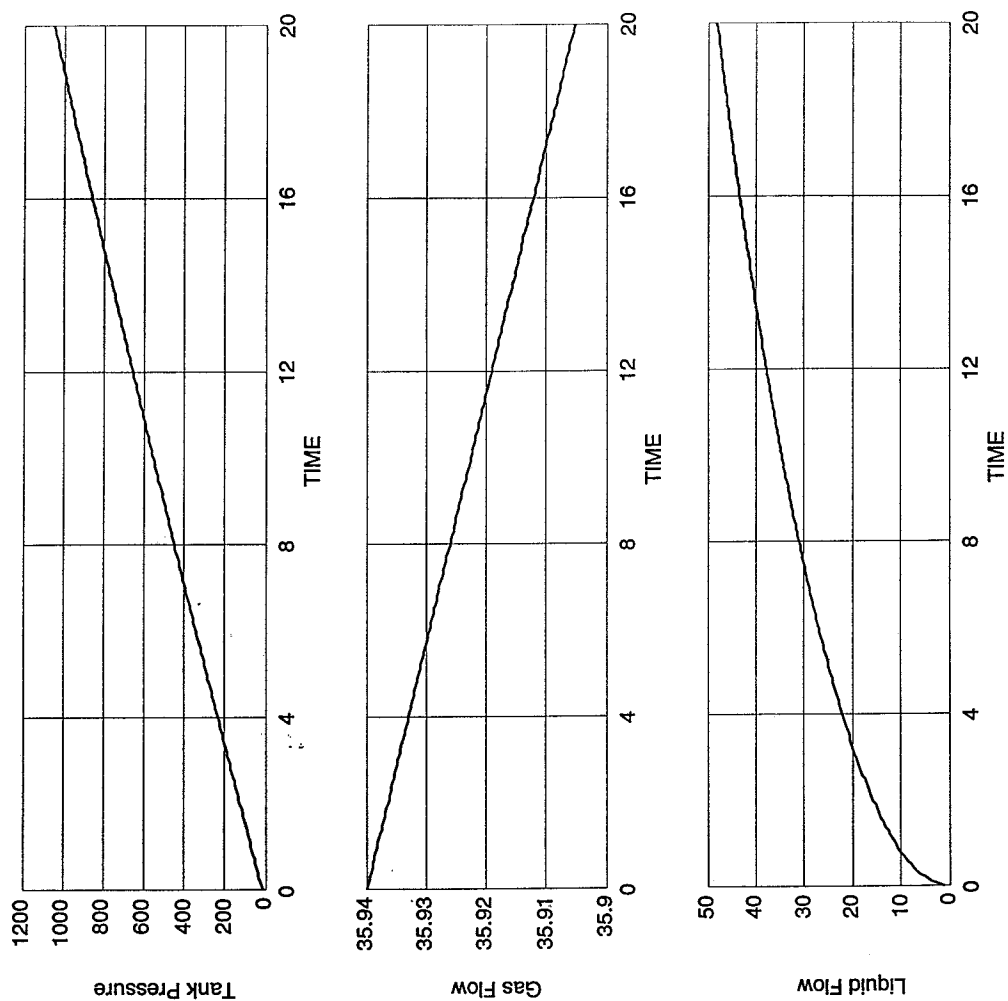


Fig. 25. Results of the Runtank Test Model Using the nr Library.

Runtank Test for er Library

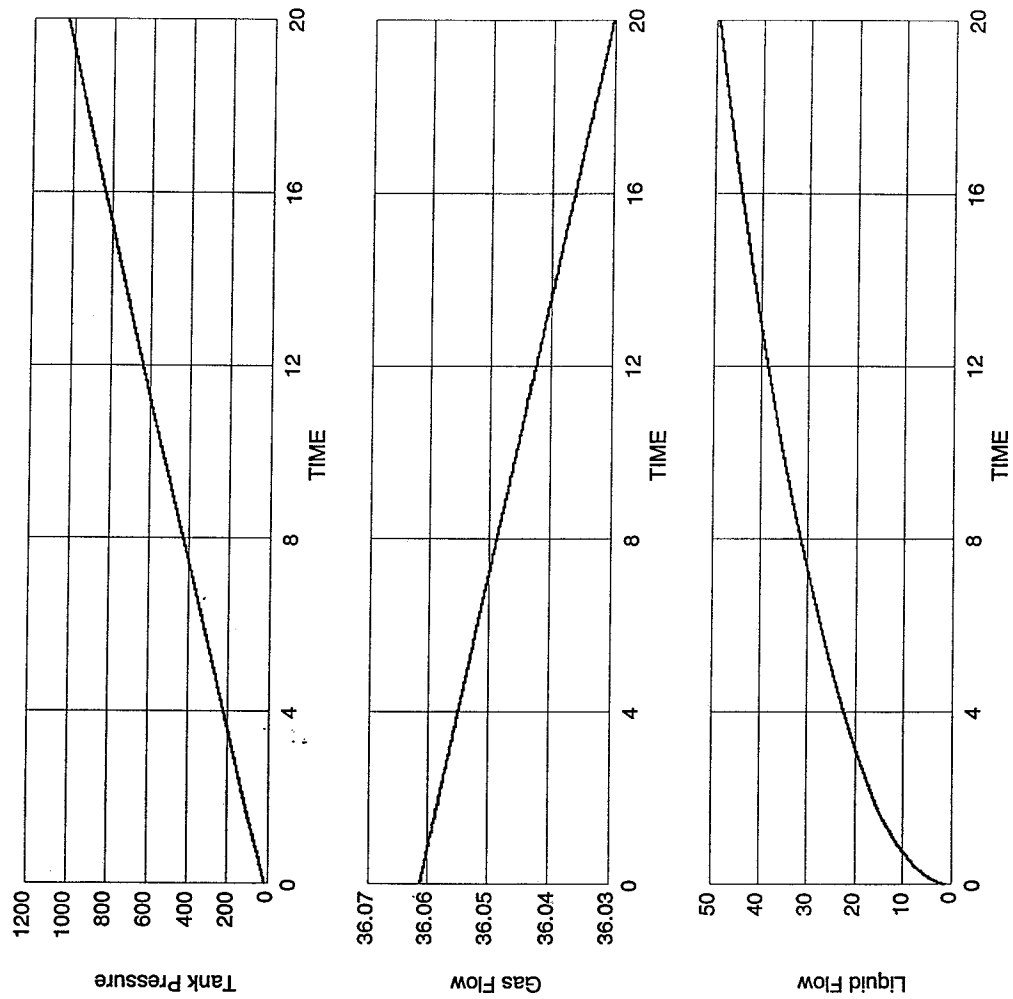


Fig. 26. Results of the Runtank Test Model Using the er Library.

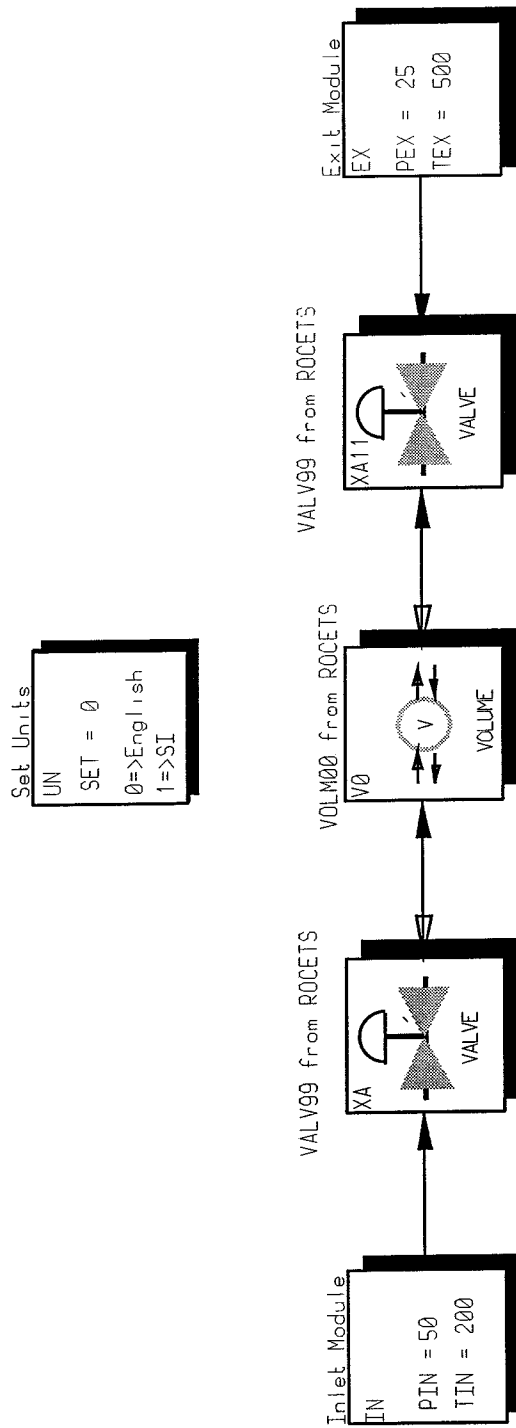


Fig. 27. A Test Model for Saturated Properties—nr Library.

N2 Saturated Properties Test--nr Library

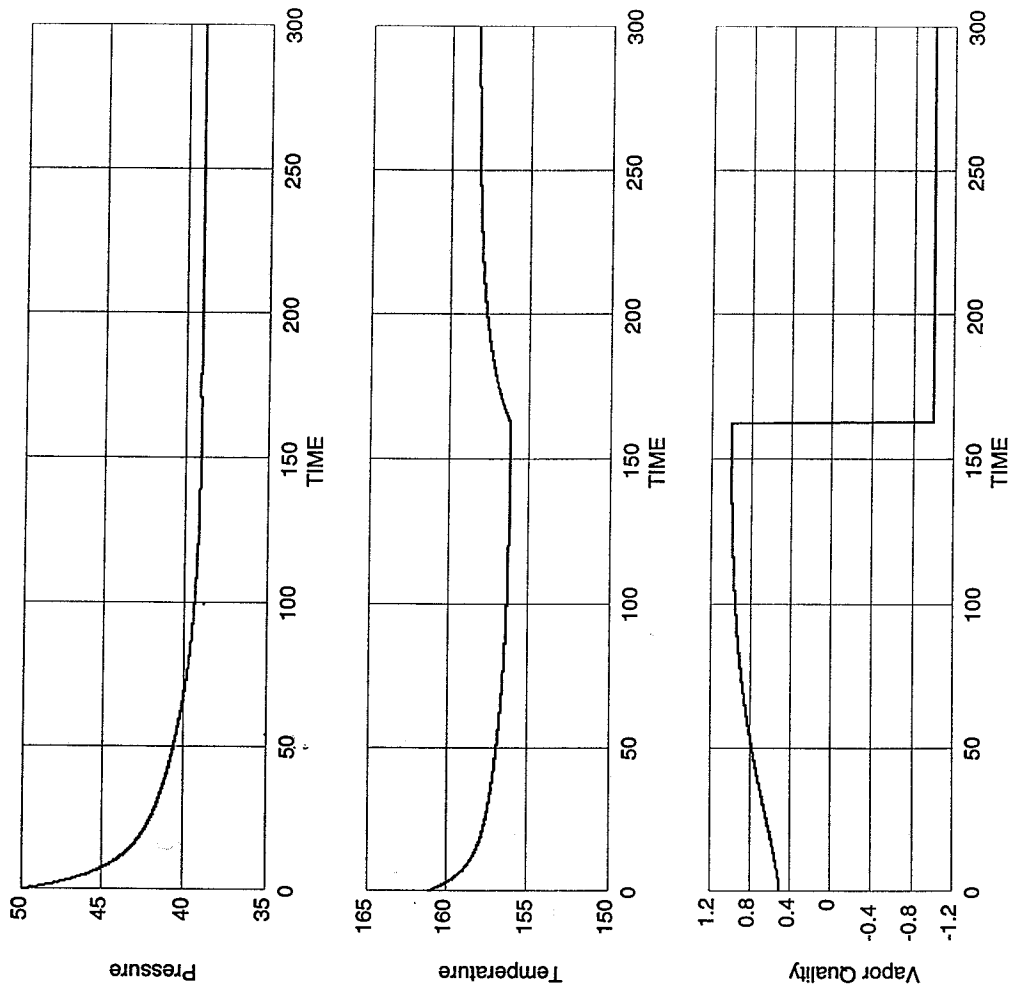


Fig. 28. Results of the Saturated Property Test Model Using the nr Library.

N2 Saturated Properties Test--nr Library

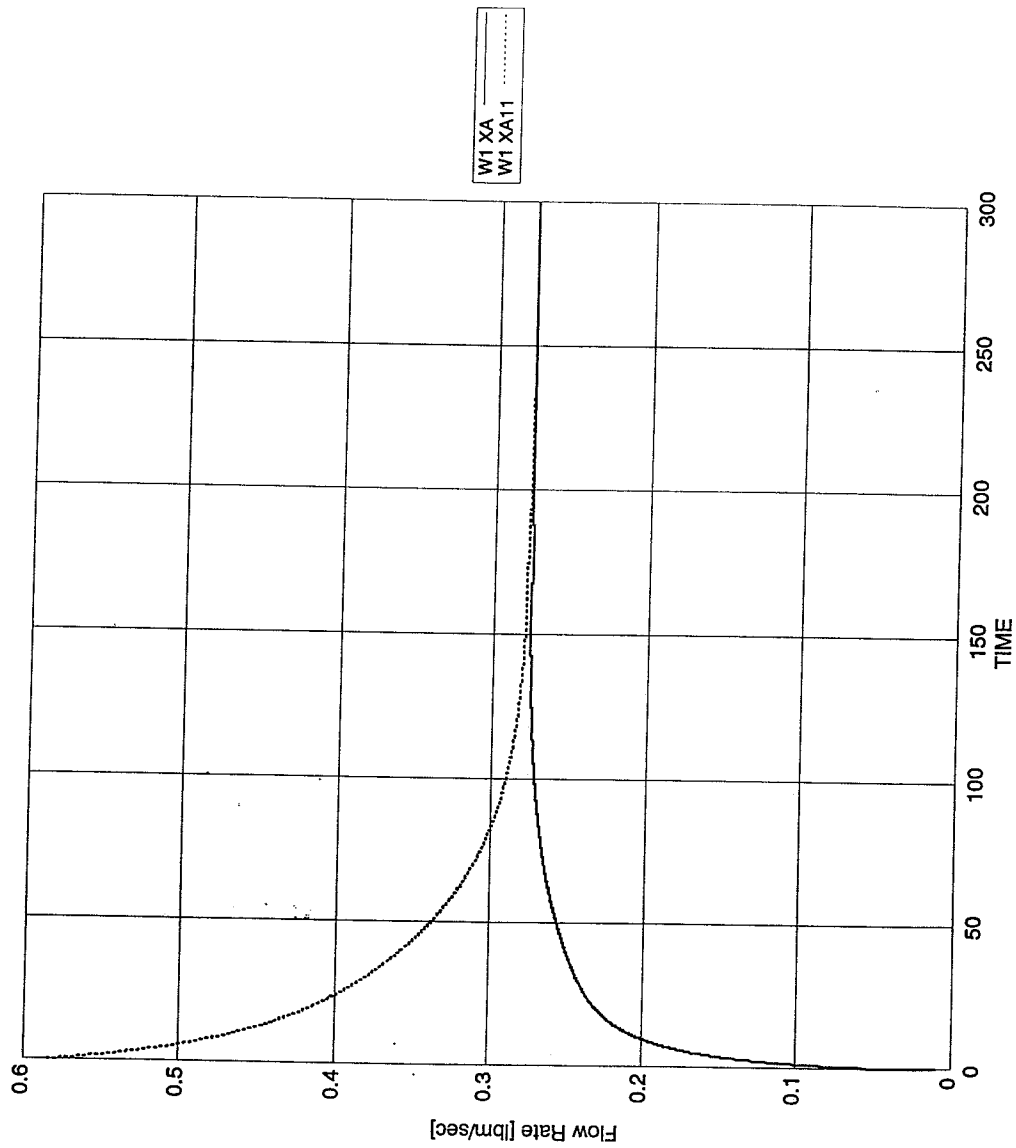


Fig. 29. Results of the Saturated Property Test Model Using the nr Library.

SECTION 3

E-1 MODEL DEVELOPMENT UPDATES

Preliminary EASY/ROCETS dynamic flow simulation models have been developed for the H₂ and O₂ flow systems on the E-1 Test Stand (formerly known as the Component Test Facility). These models have been updated based on data supplied by Mr. Larry deQuay of NASA Stennis Space Center (deQuay, 1997). These new models are discussed below.

High-Pressure Hydrogen

The high-pressure hydrogen system is shown schematically in Figure 30. High-pressure H₂ gas is supplied by the bottles. Part of this gas is used in the mixer to temper the liquid H₂ before it is fed into the test articles. Parallel paths are allowed for flow to Test Cells 2 and 3. Figure 31 shows the EASY/ROCETS model for the system prepared in the old er-library. Controls are supplied for the runtank pressure and the liquid and mixer-gas flow rates. Figure 32 shows the same model developed for the new NIST-code nr-library.

The handwritten notes for the model development are given in Appendix I.

Figures 33-35 show the results for a trial run using the er-library model. In Figure 33, the runtank pressure is seen to be held almost constant by the pressure control system. Figure 34 shows the flow rates for the runtank pressurant gas, the liquid, and the mixer gas. The mixer-gas and liquid flow rates are maintained to set schedules by the controllers. Figure 35 shows the control valve flow coefficient for the pressurant gas. The two bumps in the curve between 6 and 8 seconds are responses to scheduled bumps

in the liquid flow (see Figure 34). Figures 36-38 show the same results for the nr-library. As they should be, they are seen to be nearly identical to Figures 33-35. The small differences in the starting values are due to the difficulty of setting the two models to exactly the same initial conditions.

High-Pressure Oxygen

The high-pressure oxygen system is shown schematically in Figure 39. The gas bottles feed nitrogen gas to the runtank as a pressurant. The liquid oxygen is supplied to Test Cells 1 and 2. Figure 40 shows the EASY/ROCETS model for the system using the er-library. Figure 41 shows the equivalent model using the nr-library. The handwritten notes for the model development are given in Appendix I.

Figures 42-43 show results of a sample run using the er-library model. Duplicate runs using the nr-library model were practically identical.

Low-Pressure Oxygen

Figure 45 shows a schematic diagram of the low-pressure oxygen system. Liquid O₂ is supplied in parallel runs to Test Cells 2 and 3. The nitrogen pressurant is regulated to 2500 psi upstream of the runtank pressure control valve. Figure 46 shows the EASY/ROCETS model developed in the er-library. Figure 47 shows the same model developed in the nr-library. Figures 48-49 show results of a sample run using the nr-library model. Similar runs using the er-library were virtually identical. The handwritten notes for the model development are given in Appendix I.

Low-Pressure Hydrogen

The low-pressure hydrogen system is shown schematically in Figure 51. This system has not changed since the previous models were developed (Follett and Taylor, 1996) but models are included here for completeness. Please refer to the cited report for a discussion of the model development. Figure 52 shows the model developed using the er-library, and Figure 53 shows the same model using the nr-library. Figures 54-56 show sample runs using the nr-library model. Equivalent runs using the er-library model were practically identical.

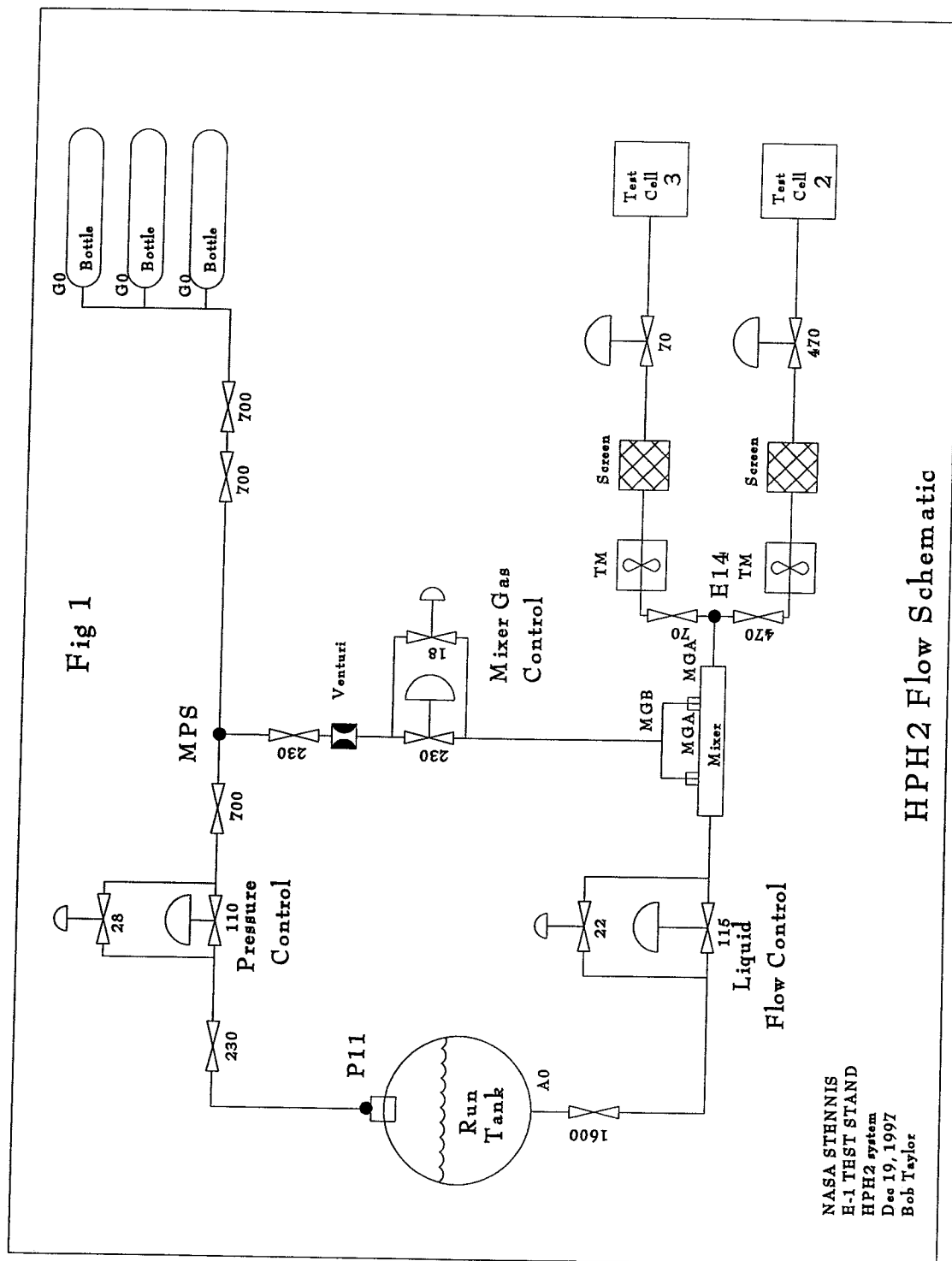


Fig. 30. Flow Schematic for the High-Pressure Hydrogen System

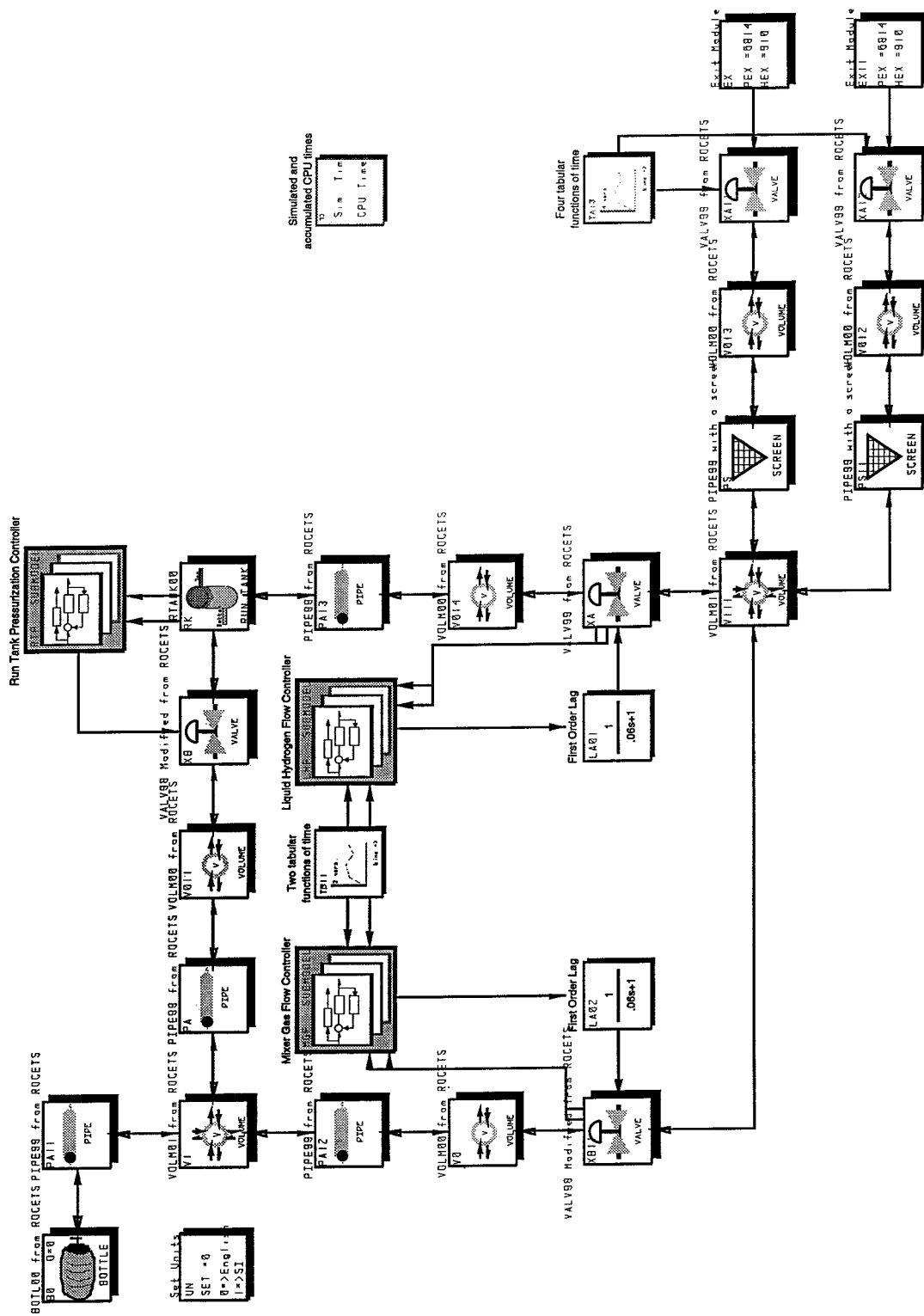


Fig. 31 EASY/ROCETS Model for the HPH2 System Using er-Library.

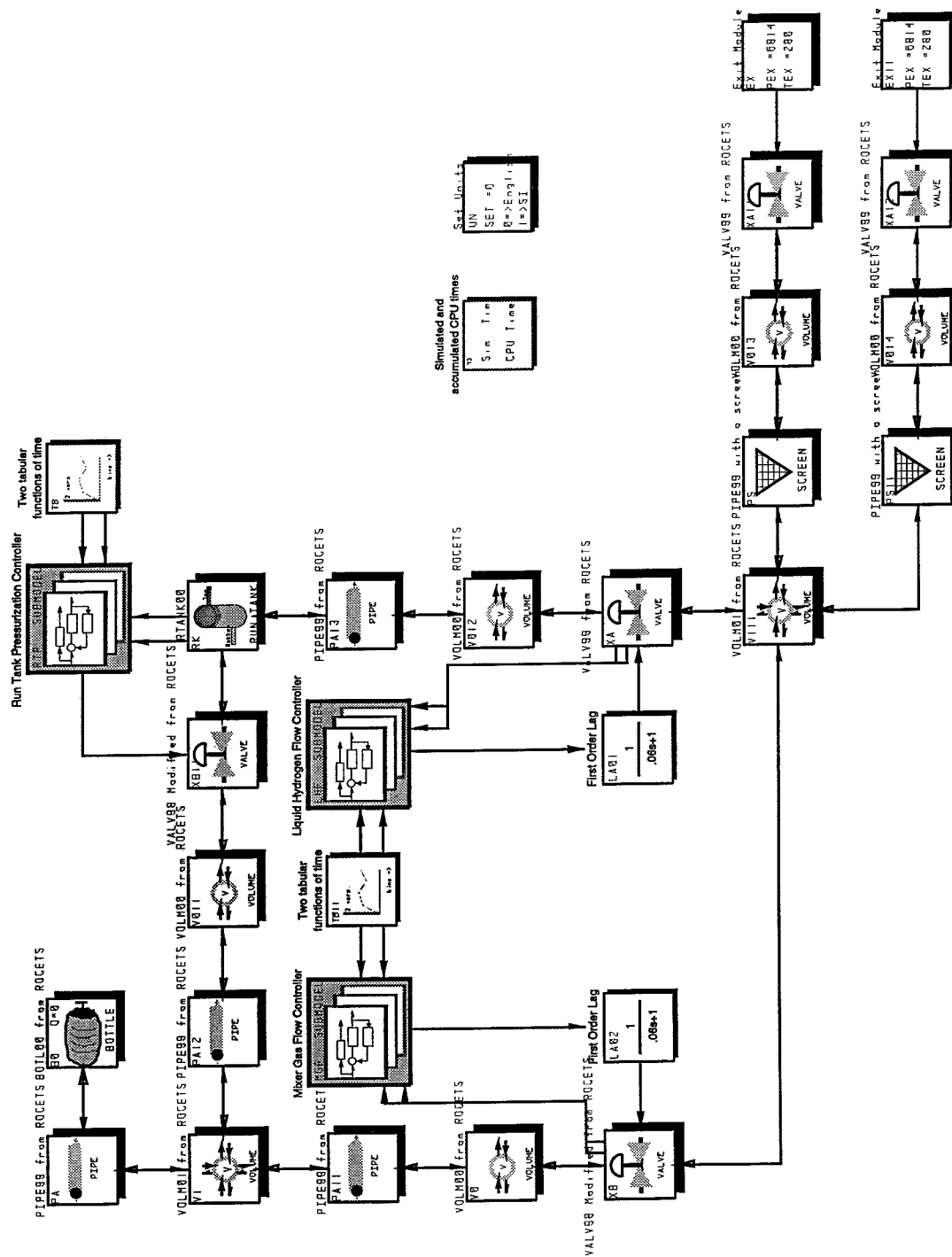


Fig. 32 EASY/ROCETS Model for the HPH2 System Using nr-Library.

High-Pressure H2 System--er Library

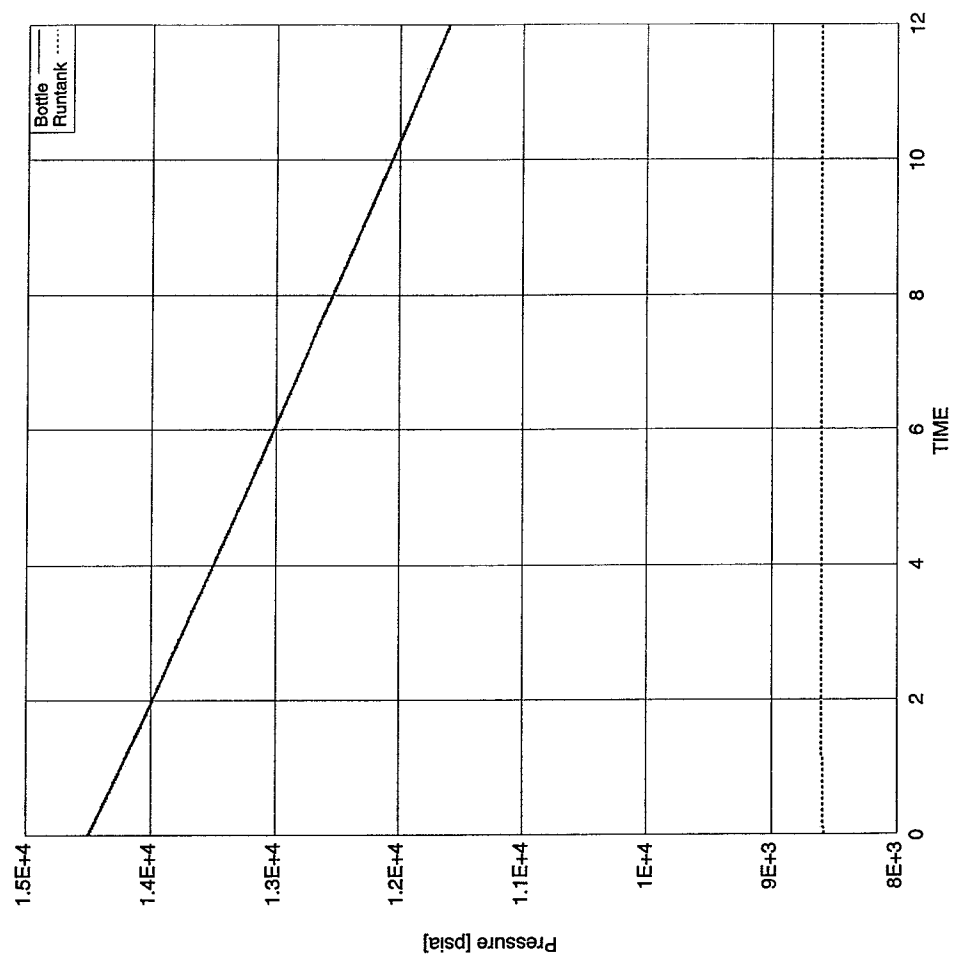


Fig. 33. Pressure versus Time Plot for HPH2 Bottle and Runtank--er-Library.

High-Pressure H2 System--er Library

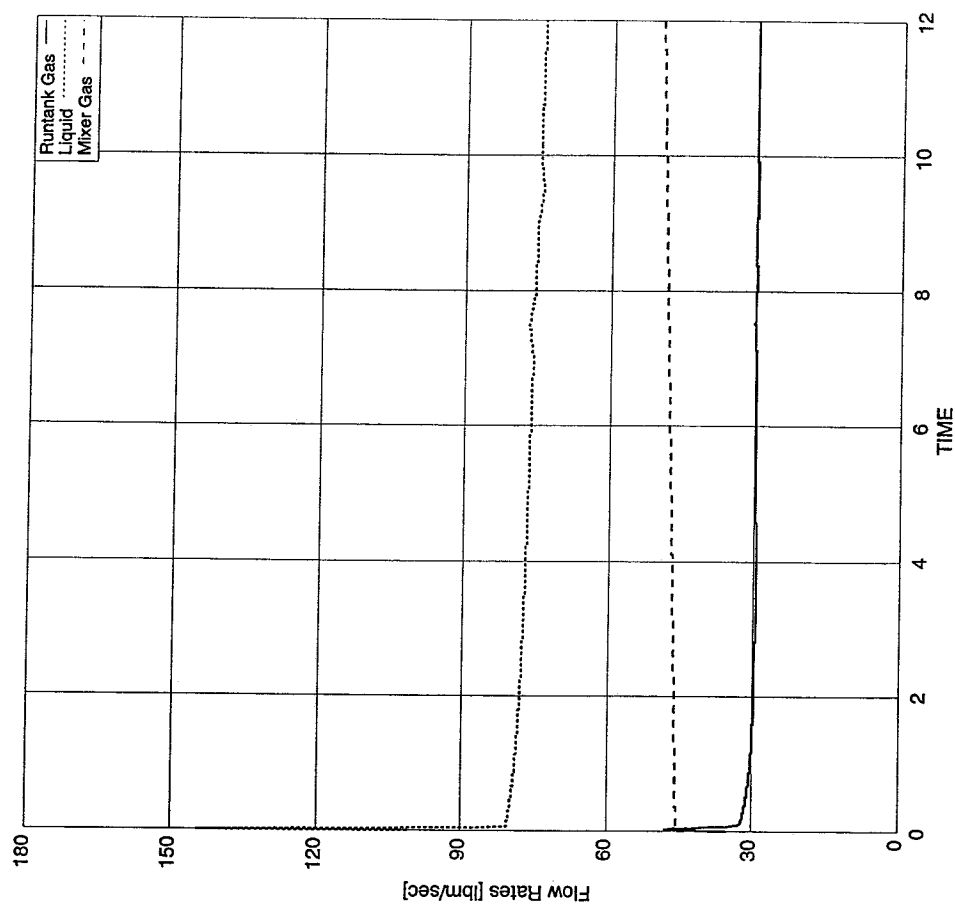


Fig. 34. Flow Rates versus Time for the Pressurant and Mixer Gas and the Liquid in the HPH2 System--er-Library.

High-Pressure H2 System--er Library

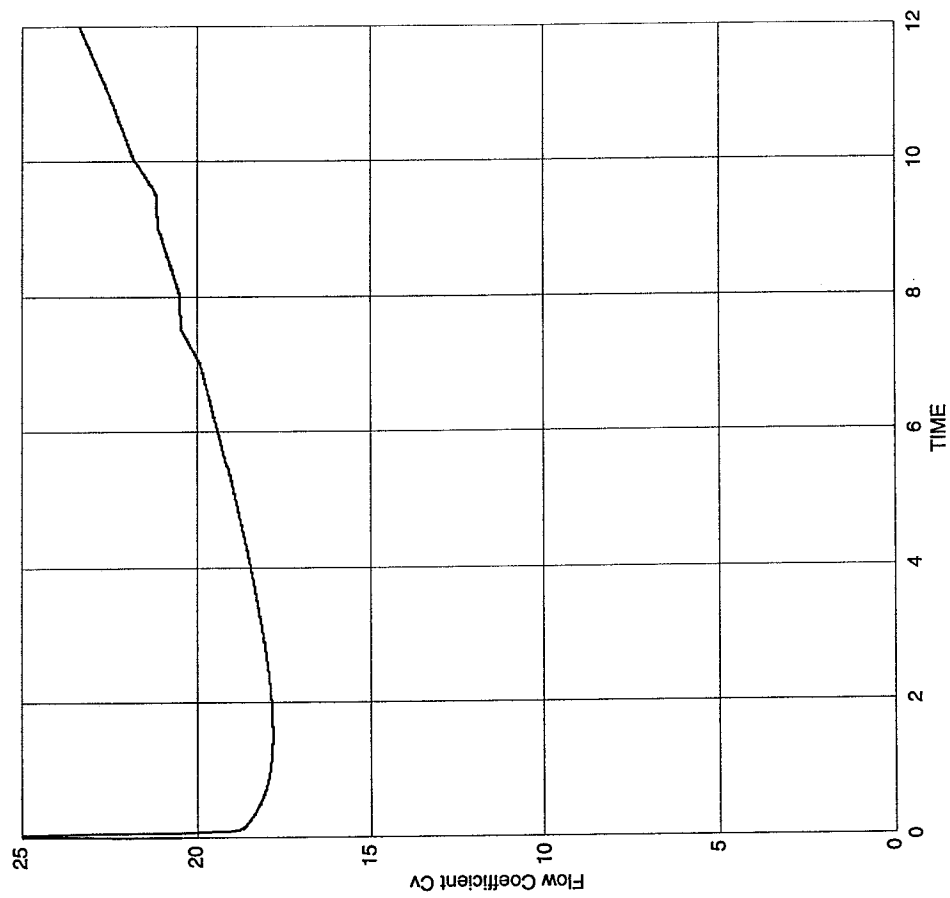


Fig. 35. Pressure Control Valve Flow Coefficient for the HPH2 System--er Library.

High-Pressure H2 System--nr Library

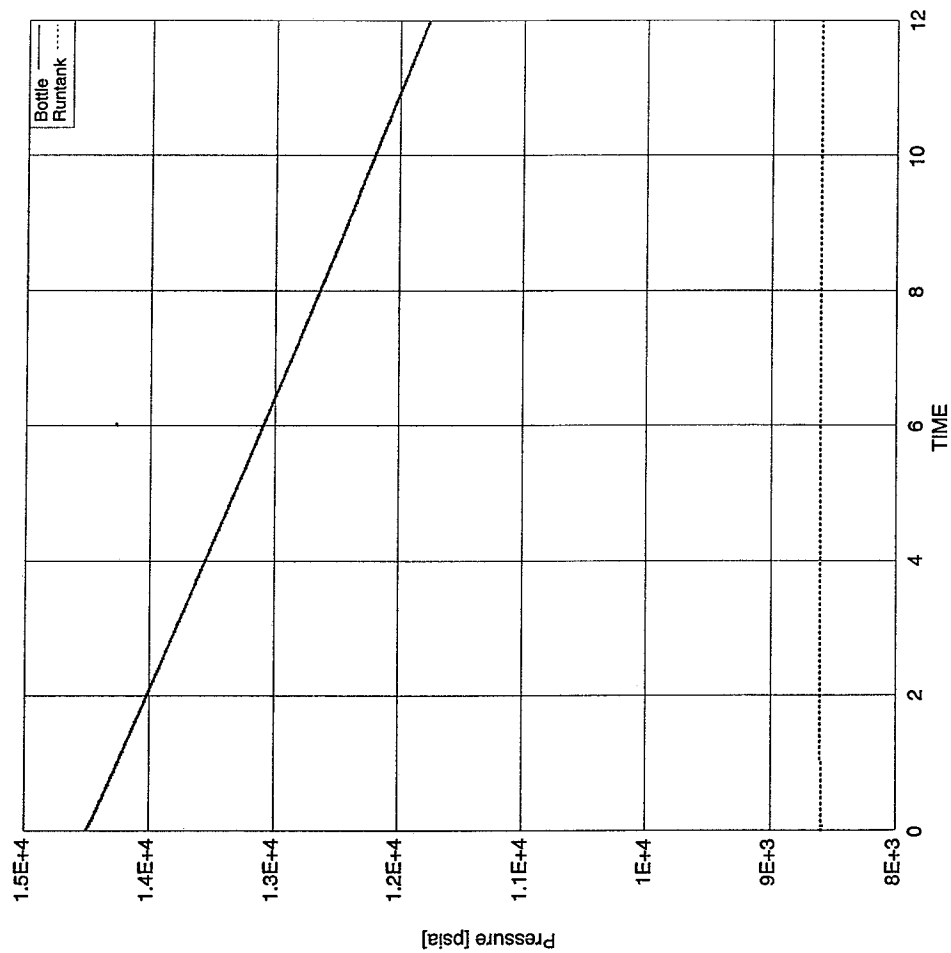


Fig. 36. Pressure versus Time Plot for HPH2 Bottle and Runtank--nr Library.

High-Pressure H2 System--nr Library

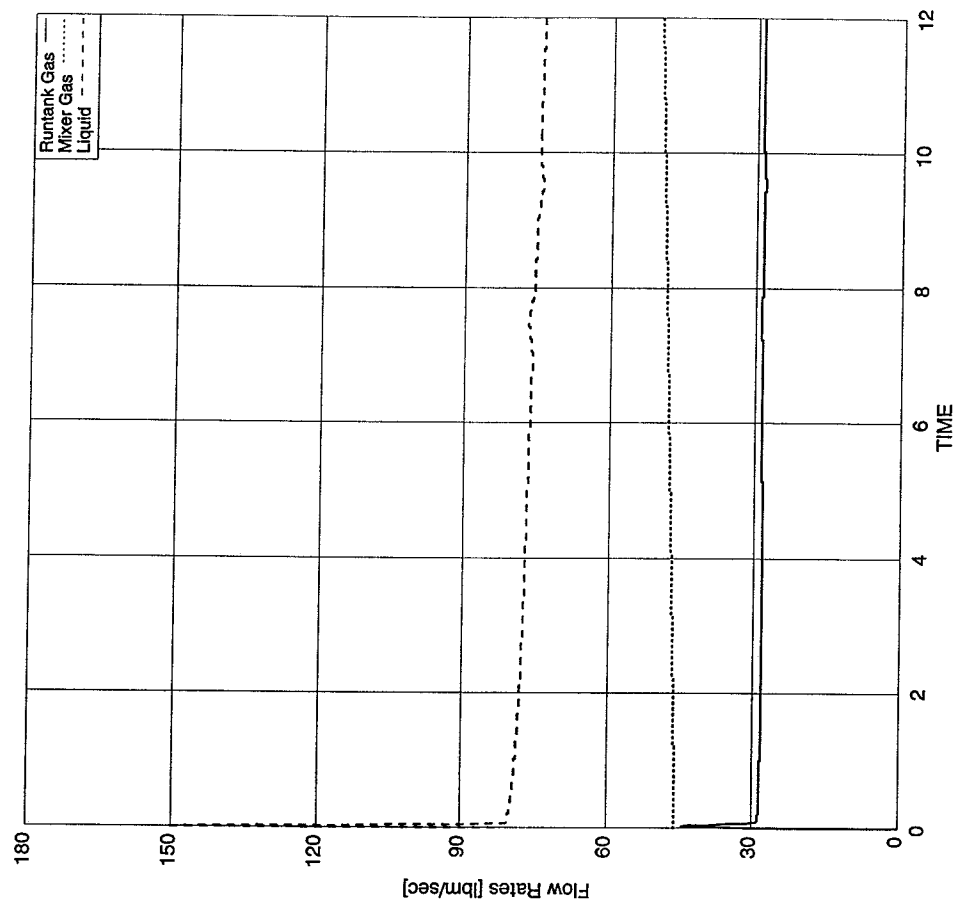


Fig. 37. Flow Rates versus Time for the Pressurant and Mixer Gas and the Liquid in the HPH2 System--nr-Library.

High-Pressure H2 System--nr Library

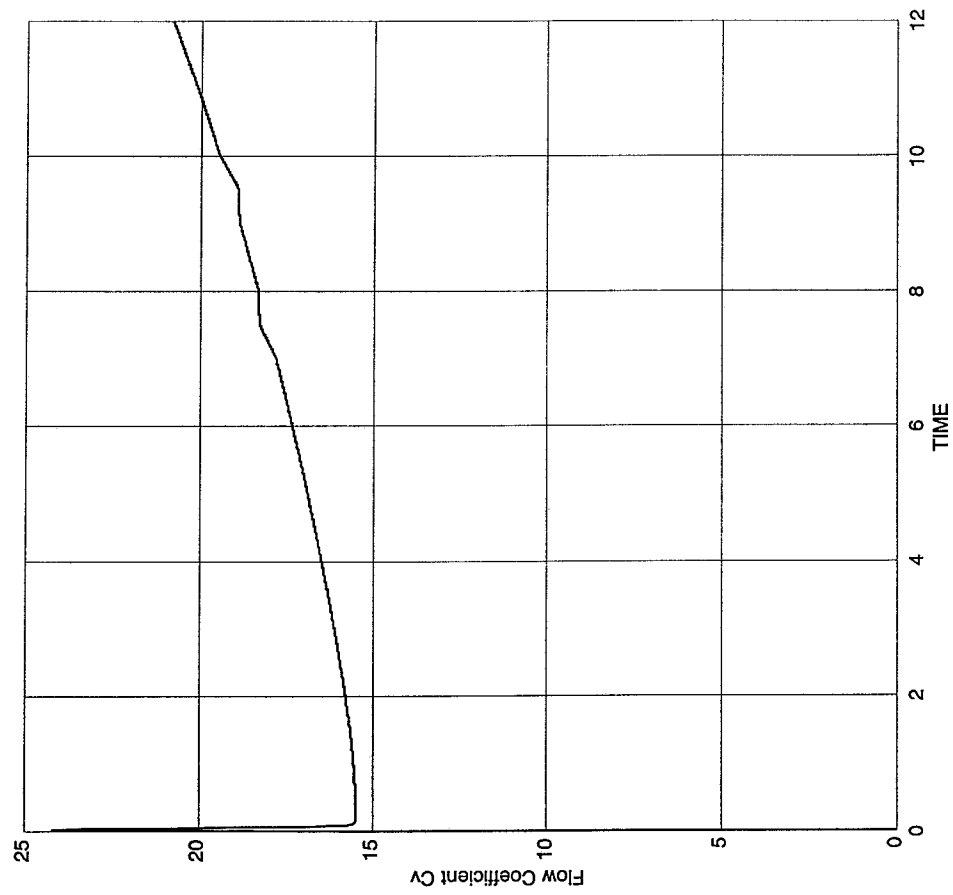


Fig. 38. Pressure Control Valve Flow Coefficient for the HPH2 System—nr-Library.

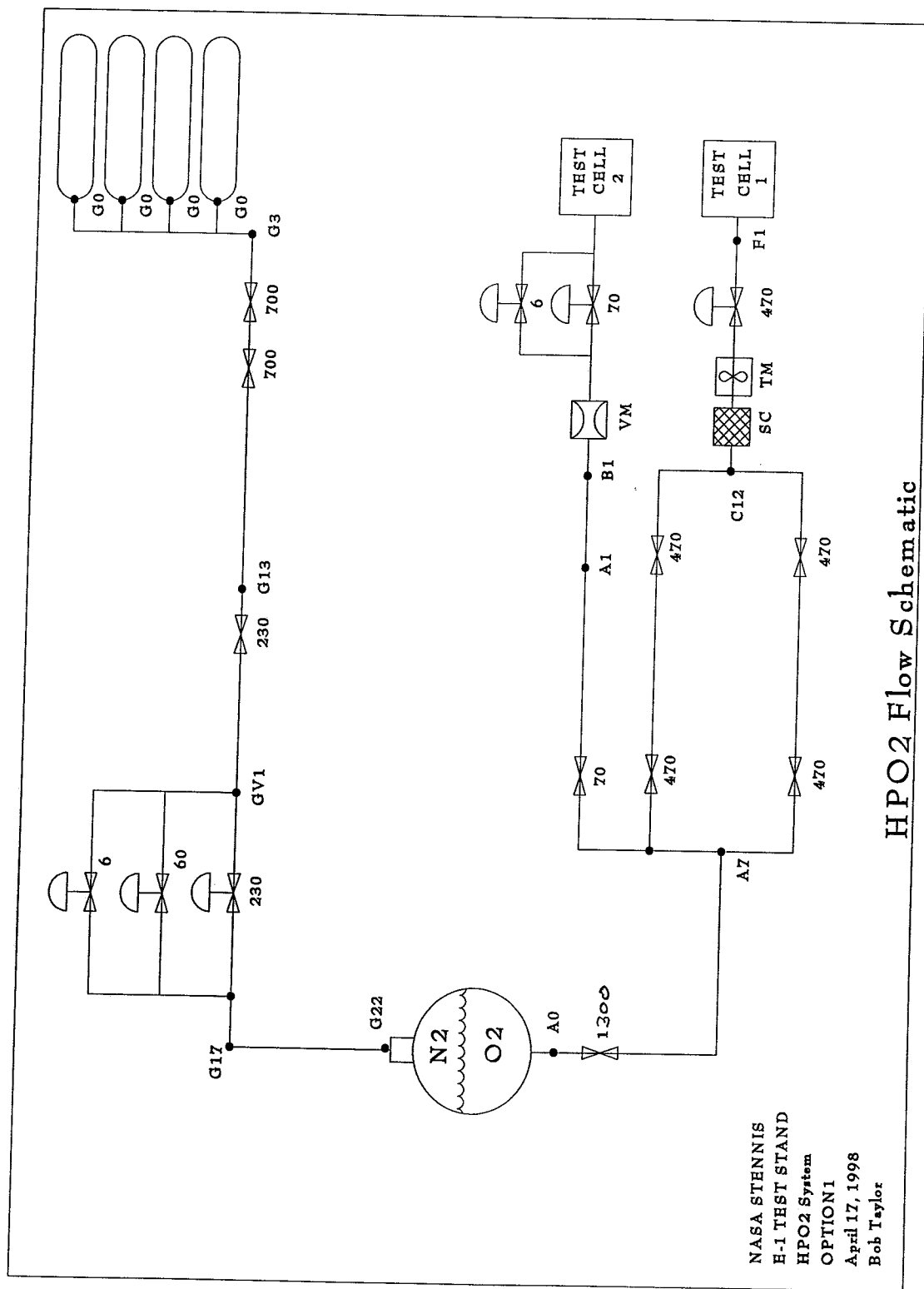


Fig. 39. Flow Schematic for the High-Pressure Oxygen System.

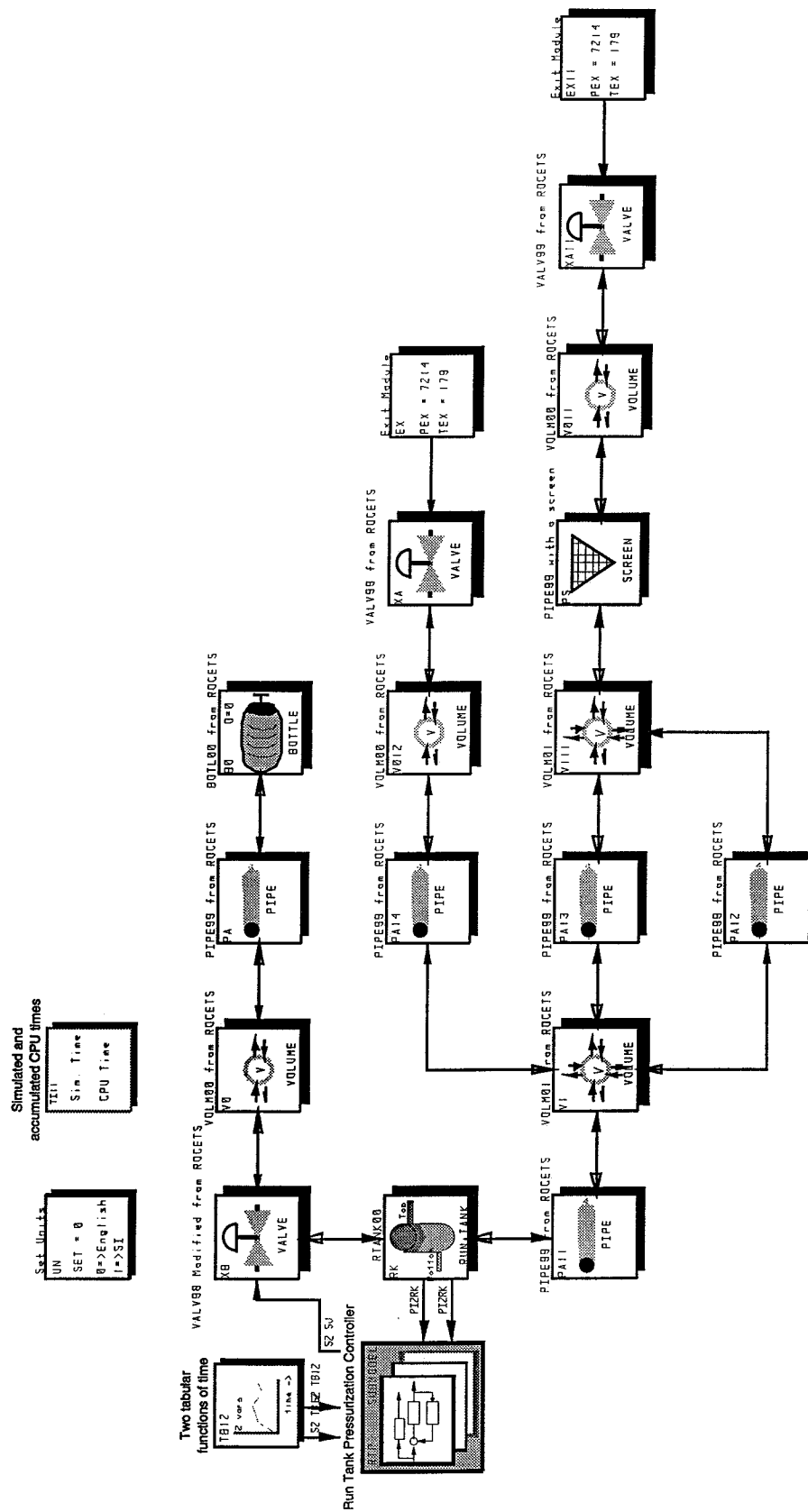


Fig. 41. EASY/ROCETS Model for HPO2 System—nr-Library.

High-Pressure O2 System--er Library

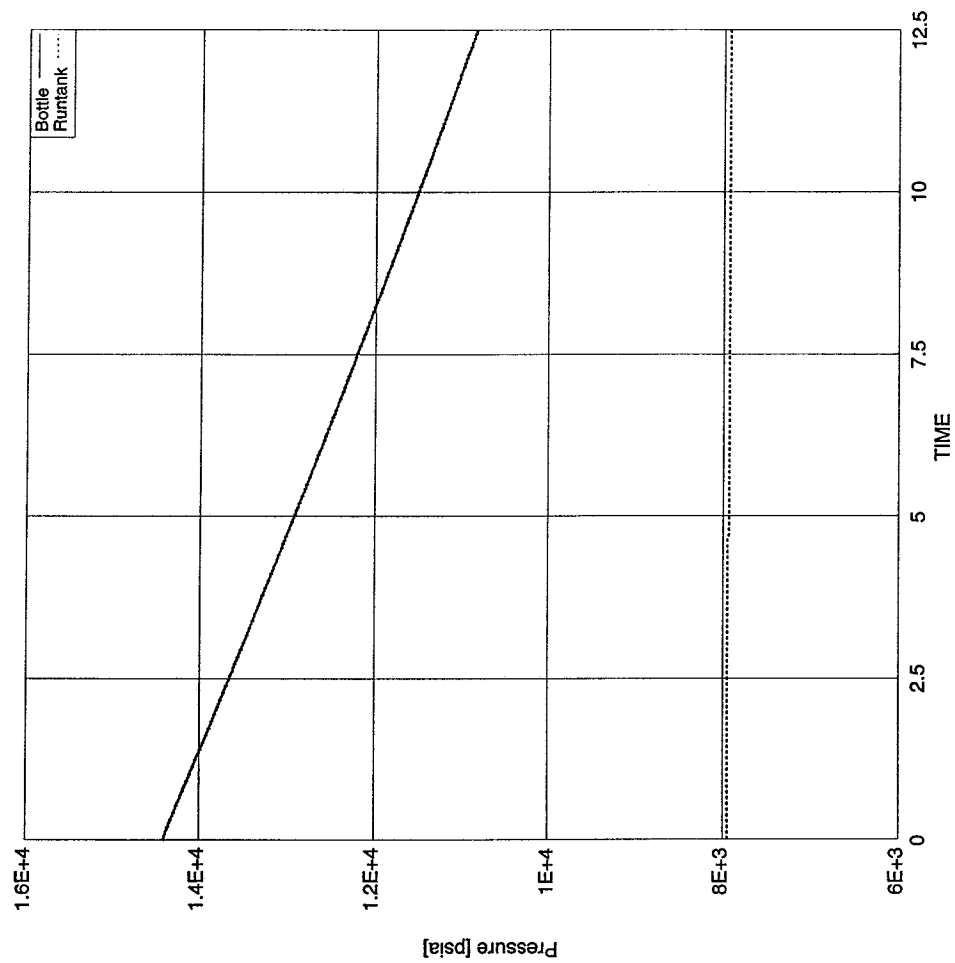


Fig. 42. Pressure versus Time for the Bottle and Runtank—er-Library.

High-Pressure O2 System--er Library

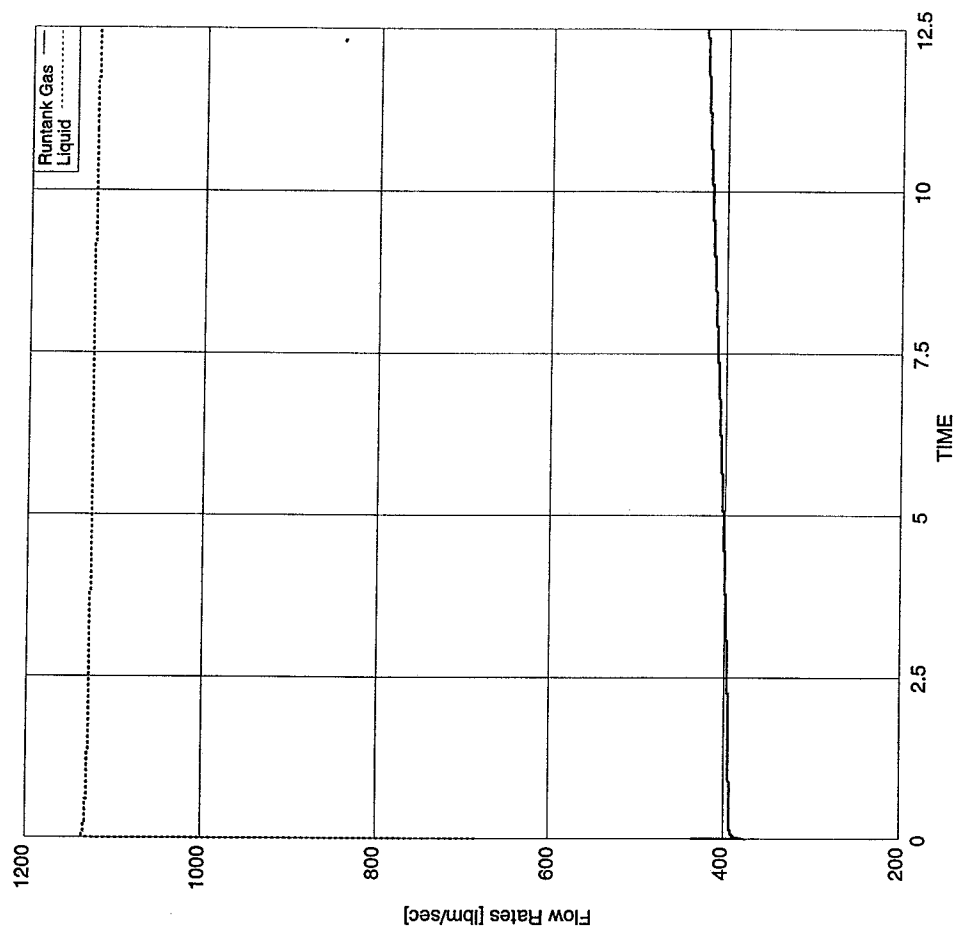


Fig. 43. Pressurant Gas and Liquid O2 Flow Rates or HPO2 System--er Library.

High-Pressure O2 System--er Library

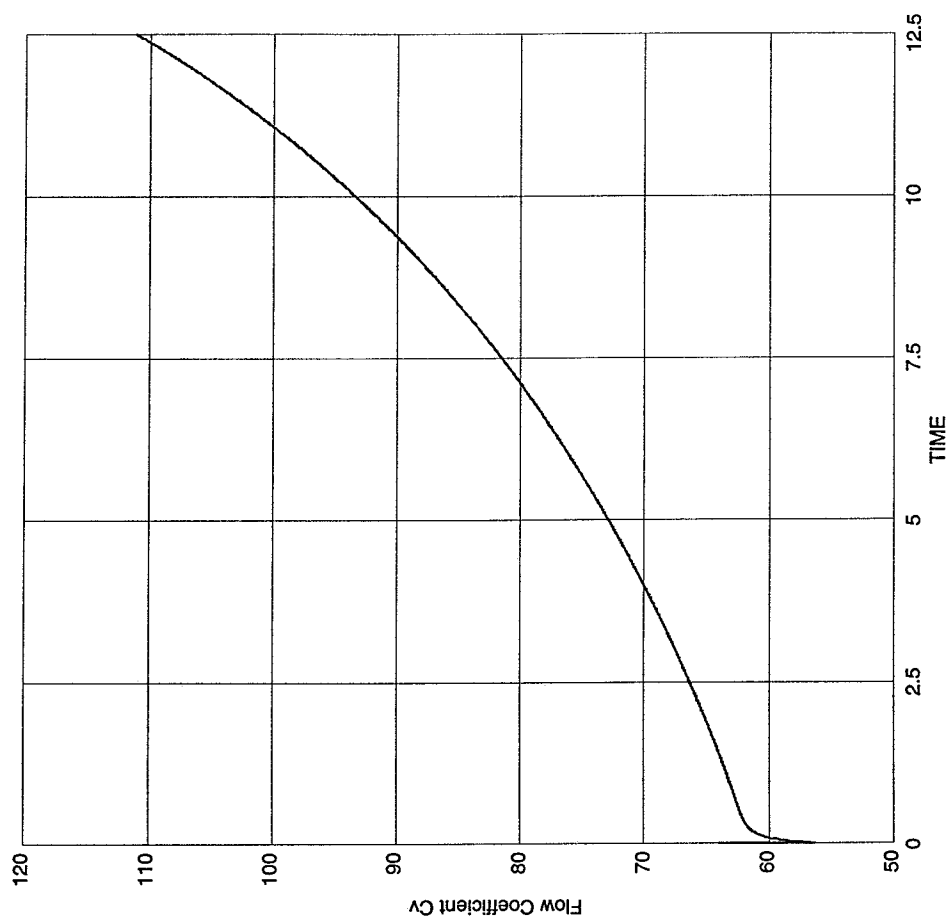
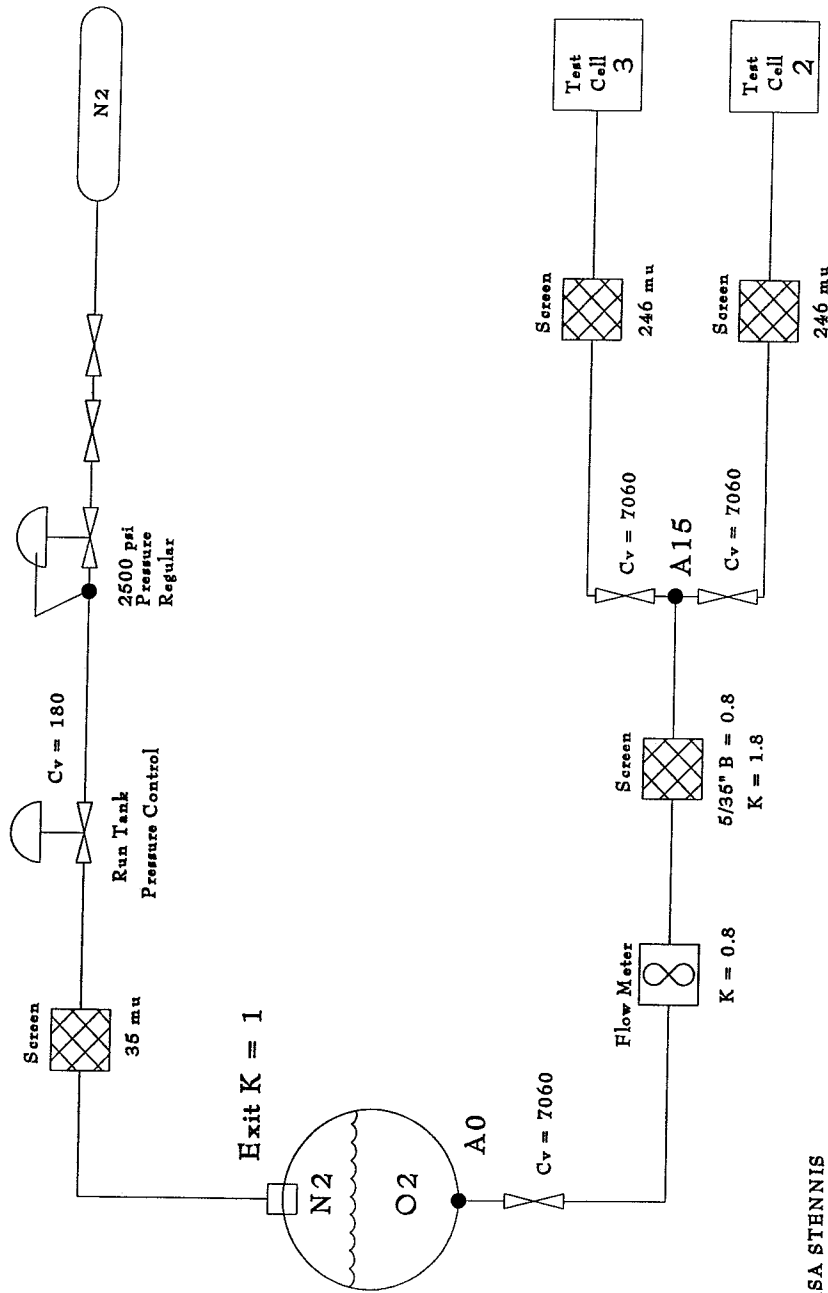


Fig. 44. Pressure Control Valve Flow Coefficient HPO2 System--er Library.

N2 SYSTEM IS UNCHANGED



NASA STENNIS
E-1 THST STAND
LPO2 System
Dec 19, 1997
Bob Taylor

LPO2 Flow Schematic

Fig. 45. Flow Schematic for the Low-Pressure Oxygen System.

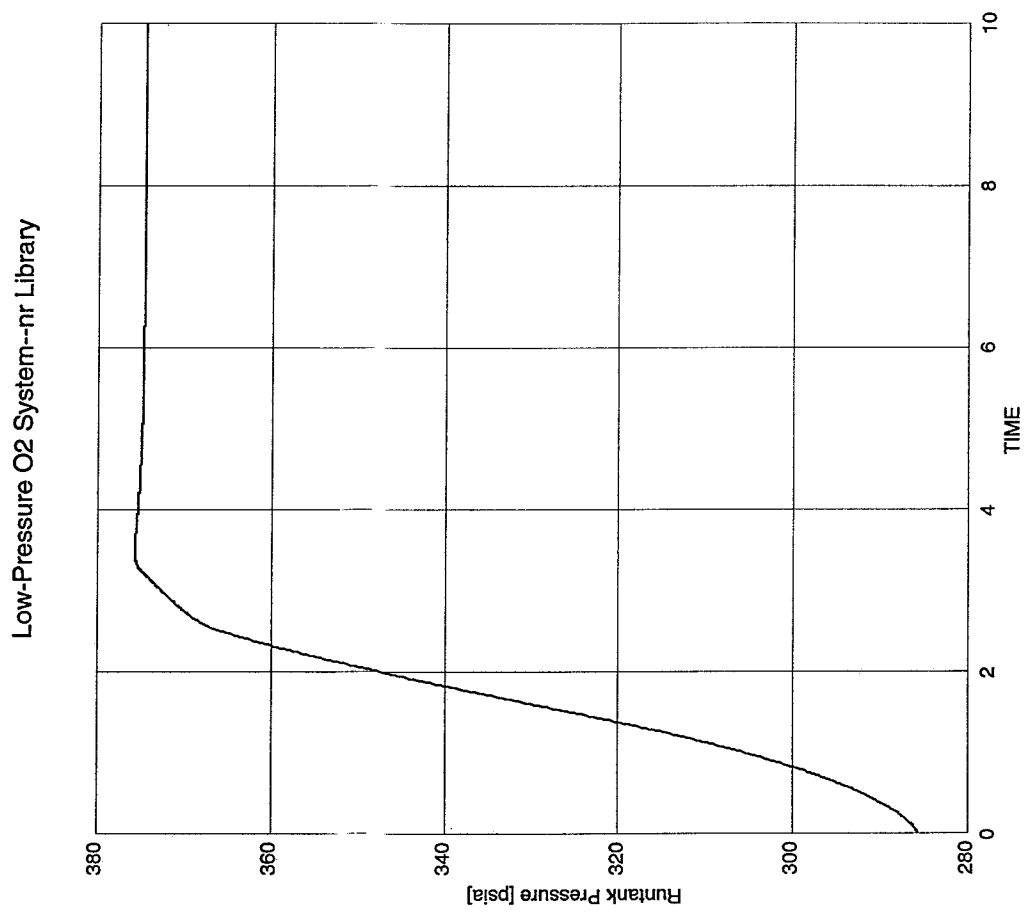


Fig. 48. Runtank Pressure History for the LPO2 System--nr-Library.

Low-Pressure O2 System--nr Library

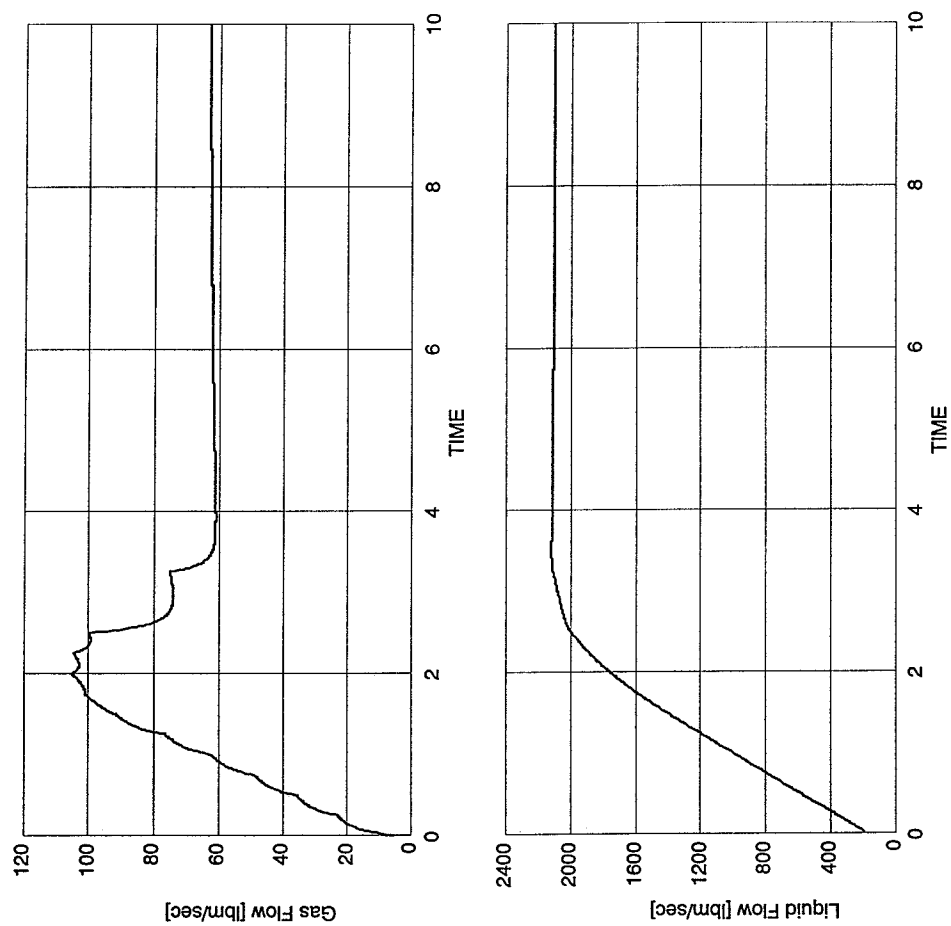


Fig. 49. Pressurant Gas and Liquid O2 Flow Rates for LPO2 System—nr-Library.

Low-Pressure O2 System--nr Library

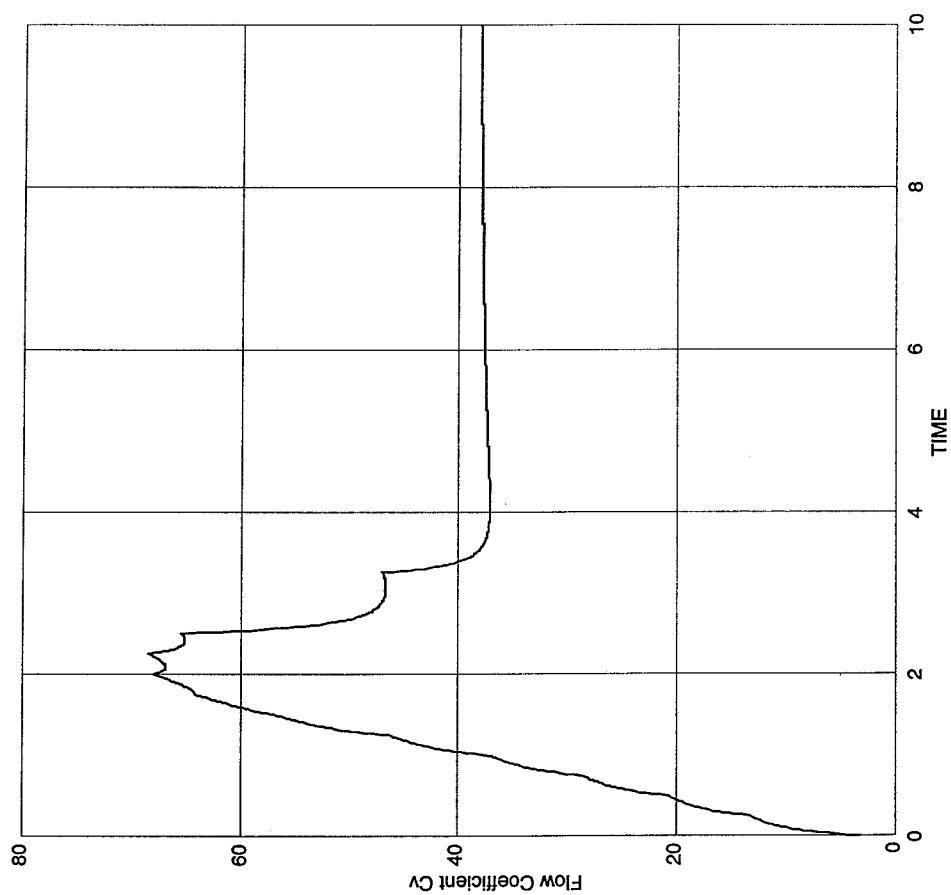


Fig. 50. Pressure Control Valve Flow Coefficient LPO2 System--nr Library.

**GENERAL SCHEMATIC FOR THE
LIQUID HYDROGEN RUN SYSTEM DFFSM
(LH-DFFSM-1.1.8 Apl)**

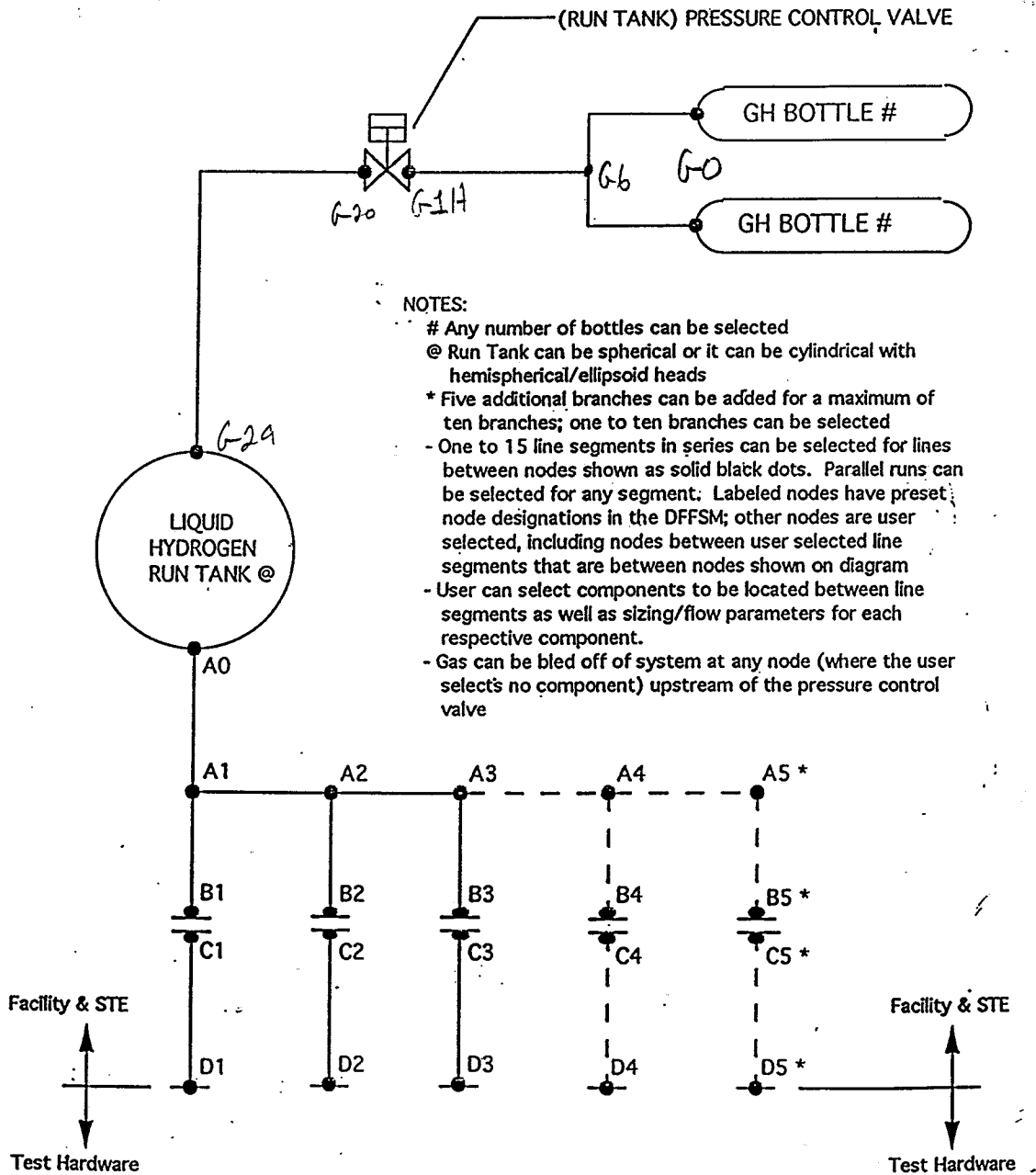


Fig. 51. Flow Schematic for the Low-Pressure Hydrogen System.

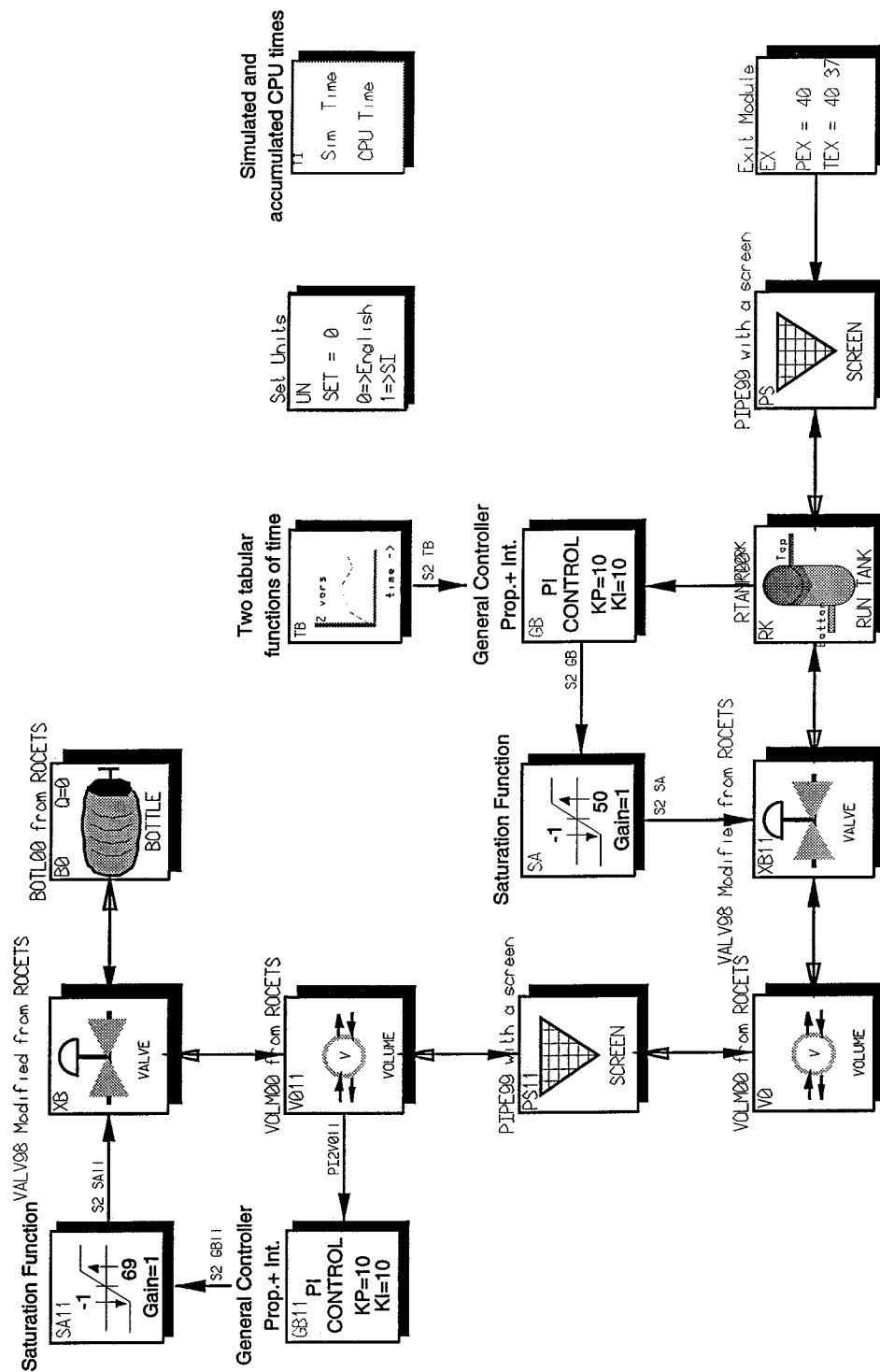


Fig. 53. EASY/ROCETS Model for LPH2 System—nr-Library.

Low-Pressure H2 System--nr Library

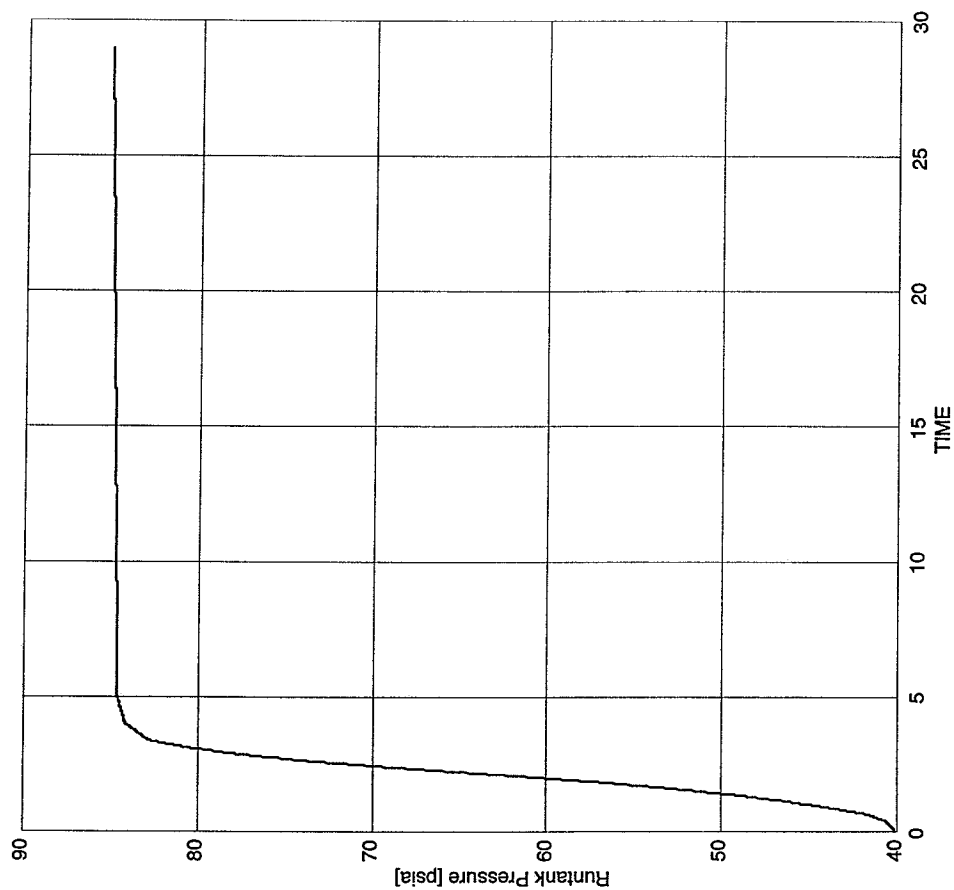


Fig. 54. Runtank Pressure History LPH2 System--nr-Library.

Low-Pressure H2 System--nr Library

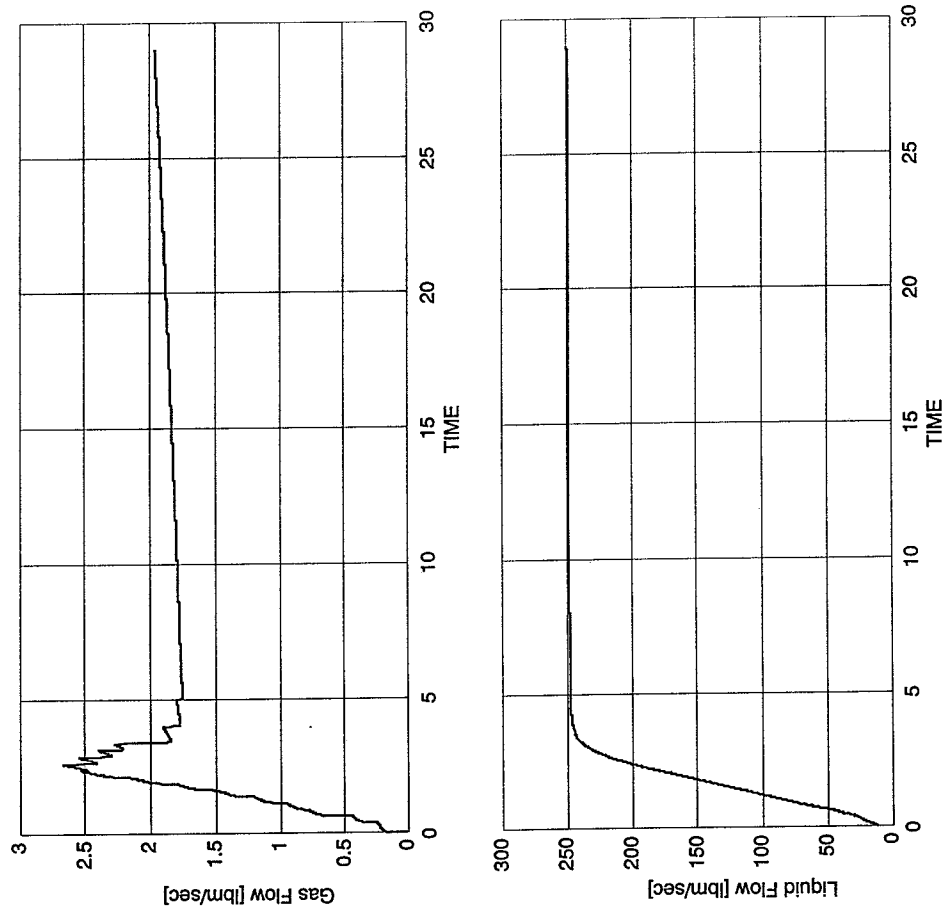


Fig. 55. Pressurant Gas and Liquid H2 Flow Rates LPH2 System—nr-Library.

Low-Pressure H2 System--nr Library

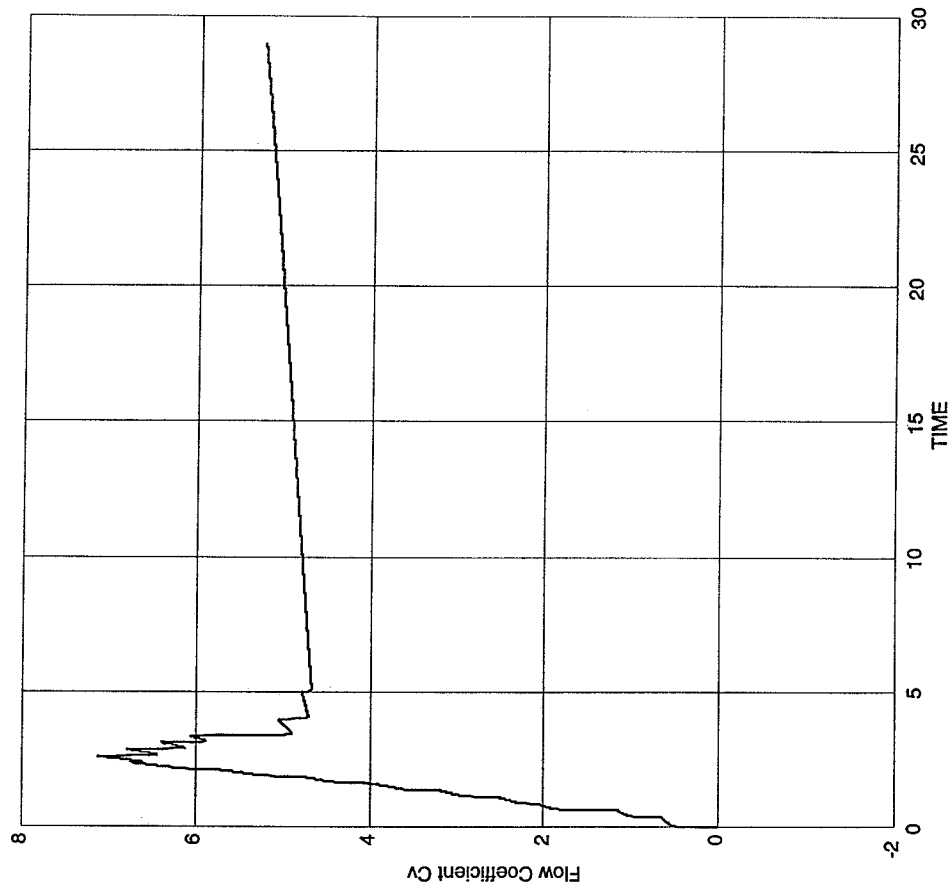


Fig. 56. Pressure Control Valve Flow Coefficient LPH2 System--nr-Library.

SECTION 4

NEW MODULE DEVELOPMENT

New models have been developed for an Allen-Bradley SLC 5/03 PID programmable logic controller and for a control valve with built in time delay on the actuation.

Control Systems Enhancements

The existing EASY/ROCETS models for CTF have used general-purpose components to implement the PID (proportional plus integral plus derivative) controllers. The actual control is planned to implemented using programmable logic controllers (PLC's). In particular, an Allen-Bradley SLC-5/03 PLC has been specified by NASA personnel as a representative device. This section details the development and use of an EASY/ROCETS module which provides a model of the PID control function of the AB SLC 5/03, which actually is constant over the SLC 5 series of PLC's.

Derivation

A PID controller generally implements some discrete approximation to the equation

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} \quad (1)$$

Implementation on a digital computer such as the PLC requires a numerical approximation to this exact equation. In addition, the derivative term must be filtered to limit the effects of high frequency noise terms. A commonly used scheme to approach this problem is to incorporate a

low-pass filter on the derivative term [Phillips and Harbor, 1996]. If Equation (1) is converted to the Laplace domain, we get

$$CV(s) = K_p E(s) + K_I \frac{1}{s} E(s) + K_D s E(s) \quad (2)$$

where the term $(1/s)$ represents an integration and the term (s) represents a differentiation effect.

Applying the low-pass filter to Equation (2) yields

$$CV(s) = K_p E(s) + K_I \frac{1}{s} E(s) + K_D \frac{s}{1 + s/\omega_f} E(s) \quad (3)$$

The variable ω_f represents the cutoff frequency of the low-pass filter, which has transfer function

$$G_f(s) = \frac{1}{1 + s/\omega_f} \quad (4)$$

The choice of ω_f is dependent on the desired frequency response of the PID. If ω_f is chosen too small, the derivative term will be ineffective due to the rolloff of the filter $G_f(s)$. If it is chosen too large, high frequency noise will cause problems due to the amplification effect of the derivative term.

The Allen-Bradley SLC 5 series of PLC's implements a filtered form of the PID equation as described in Equation (3). However, the details for the implementation, including the placement of the derivative term filter cutoff frequency, are considered proprietary information by Allen-Bradley personnel. To date, no clear information regarding the details of the implementation have been forthcoming.

In order to provide a component for use in the EASY/ROCETS environment, certain assumptions have been made regarding the detailed operation of the PLC version of the PID controller. First, it is assumed that the calculations are simply discretized state models of the

integral and derivative terms of the control variable as shown in Equation (3). Additionally, the filter frequency ω_f is placed at the frequency of operation of the PID, which is specified as the scan time of the ladder diagram. Note that this is a reasonable assumption based on sampling theory. It is well-known that the sample rate should be twice the highest frequency of interest. By filtering above this, we tend to limit the high-frequency terms that are considered to be noise. It should also be noted that this choice was made after observation of the operation of an example system with a variety of scan times and gain constants.

By splitting Equation (3) into three pieces, we can calculate the proportional term, the integral term, and the derivative term individually, then combine them to give the complete answer for the control variable. The three terms are given in Equation (5):

$$\begin{aligned} CV_P(s) &= K_P E(s) \\ CV_I(s) &= \frac{K_I}{s} E(s) \\ CV_D(s) &= \frac{K_D \omega_f}{s + \omega_f} E(s) \end{aligned} \tag{5}$$

Writing continuous state equations from these terms, using the integral and derivative terms CV_I and CV_D as states, we obtain:

$$\begin{aligned} \dot{CV}_I &= K_I E \\ \dot{CV}_D &= -\omega_f CV_D - K_D \omega_f^2 E \end{aligned} \tag{6}$$

with

$$CV = CV_I + CV_D + K_P E \tag{7}$$

Equations (6) show the state equations and Equation (7) gives the output equation. Next, the continuous equations are discretized, yielding

$$\begin{aligned}
CV_I(k+1) &= [1]CV_I(k) + [K_I \tau]E(k) \\
CV_D(k+1) &= [e^{-\omega_f \tau}]CV_D(k) + (-\omega_f K_D)(1 - e^{-\omega_f \tau})E(k)
\end{aligned} \tag{8}$$

Note that the sampling period for the discretization is represented as τ , and corresponds to the scan time of the PID controller. Using the previously mentioned assumption that $\omega_f = 1/\tau$, we get

$$\begin{aligned}
CV_I(k+1) &= [1]CV_I(k) + [K_I \tau]E(k) \\
CV_D(k+1) &= [e^{-1}]CV_D(k) + (-K_D / \tau)(1 - e^{-1})E(k)
\end{aligned} \tag{9}$$

The output equation for the system has the same form as the continuous output equation, so that

$$CV(k) = CV_I(k) + CV_D(k) + K_P E(k) \tag{10}$$

Equations (9) and (10) give the discrete-time state-space description of the operation of a standard PID controller. Customization of the PID for the AB-SLC 5 series of PLC's is accomplished by means of a conversion to a set of dependent gains, which provide an alternate description of a PID controller. For the SLC 5, the three terms which specify the operation of the PID controller are K_c (proportional gain term), T_i (integral gain or reset term), and T_d (derivative gain or rate term). Conversion between the two sets of gains is as follows:

$$\begin{aligned}
K_P &= K_c \\
K_I &= \frac{K_c}{60T_i} \\
K_D &= 60K_c T_d
\end{aligned} \tag{11}$$

where T_i has units of repeats/minute and T_d has units of minutes. Additionally, the values are entered into the SLC 5 as integer values only, so a range and gain enhancement specification allows K_c and T_i to be entered using a multiplier of either 10 or 100. T_d , however, is always entered using a multiplier of 100. The conversions can be represented using the variable "gain" to show the adjustable gain (either 10 or 100).

$$\begin{aligned}
 K_P &= \frac{K_c}{\text{gain}} \\
 K_I &= \frac{K_c}{60T_i} \\
 K_D &= \left(\frac{K_c}{\text{gain}} \right) \left(\frac{T_d}{100} \right) (60) = 0.6 \frac{K_c T_d}{\text{gain}}
 \end{aligned}
 \tag{12}$$

So, by using the conversions in Equation (12), we can implement the state and output equations of Equations (9) and (10). Component AB of the EASY/ROCETS library is an implementation of these equations. Figure 57 shows the icon for the AB component, and Figure 58 shows a portion of the Input Specification Table.

AB-SLC-5/03 PID

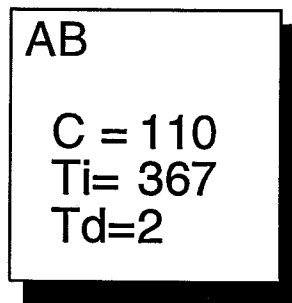


Fig. 57. Icon for Module AB (Allen-Bradley SLC-5/03)

Component Data Table												
Options												
Component AB (ab) - AB-SI-C-BB3 PID (Special)												
Inputs				Output States				Output Variables				
Name	P	Value	Name	P	IC Value	Prebit	Error	L	Name	P	Size	L
SP		019.2	ER	B		NO	0.00100	N				
RG	1.0		CVI	B		NO	0.00100	N				
C	110.0		CVI	B		NO	0.00100	N				
TI	357.0		CV	Z B		NO	0.00100	N				
TB	2.0		PVP	B		NO	0.00100	N				
TAU	0.01		CVI	B		NO	0.00100	N				
PV	1	52.01	ACV	B		NO	0.00100	N				
SC	1.0											
AP	019.2											
APD	B											

OK
Info
Cancel

Fig. 58. Component Data Table for Module AB

EASY/ROCETS Usage of AB

The principal design objective for the AB module was to develop a model that has I/O characteristics that are as close as is possible to those of the actual device. With that in mind, the inputs for the AB module are as shown in Table 10.

Variable	Description	Units	Default Value
SP	setpoint	Engineering units or scaled for PID	1.0
RG	reset and gain range enhancement bit	unitless	1.0
C	controller gain	unitless (/10 or /100)	.99999
TI	reset term	minutes/reset (/10 or /100)	.99999
TD	rate term	minutes (/100)	.99999
TAU	loop update time	seconds	1.0
PV	process variable	Engineering units or scaled for PID	.99999
SC	scale setpoint flag	unitless	0
AP	scaling rate	unitless	1.0
APO	scaling offset	unitless	1.0
BI	bias	bias or feedforward term	0
DB	deadband	Engineering units or scaled for PID	0

Table 10. Inputs for AB PID Module in EASY/ROCETS

The variables defined in Table 1 are given default values as shown in the table. Also, the units for each variable are shown where appropriate. For SP, PV, and DB, the units can either be in engineering units, or in units scaled for the PID calculations. It should be noted that whether the variables are in scaled form or engineering units, the same numerical accuracy applies. Where default values do not make sense, the value .99999 is used. Note that if SC is set to 0, the

values of the scaling constants AP and APO do not matter. The primary output of the module is CV, the control variable output state. Other output states provided include the current error (ER), the control variable contributions due to the integral and derivative terms (CVI and CVD, respectively), and the PID versions of the process variable and the control variable (PVP and CVP, respectively). In addition, there is a state ACV that is the control variable without bias. It is an intermediate variable used in the calculation of the eventual output CV.

Demonstration of AB Usage

To demonstrate the use of the new AB module, a simple system is shown in Figure 59. The system is a third-order transfer function with poles at 0, -1, and -2. It is desired to maintain a constant setpoint of 1, using PID control. This system has been analyzed using the actual PLC hardware as well as in the EASY/ROCETS environment. For the hardware setup, the dynamic system was simulated in real-time using a personal computer running VisSim/32 and an I/O card to provide real-time input and output.

The SLC-5/03 which was provided by NASA personnel for testing uses a current loop output module, which requires a conversion to a voltage signal in order to work properly with the VisSim simulation of the system plant. A 500 Ω resistance was used, which produced a 10V output for the maximum output of 20 mA. In order to obtain a bipolar output, a shift of -5V was included inside the VisSim simulation. This setup is shown in Figure 60. Note that the control variable and the process variable are shown in their various configurations as CV and PV. The simulation diagram of the EASY/ROCETS version of the system is shown in Figure 61.

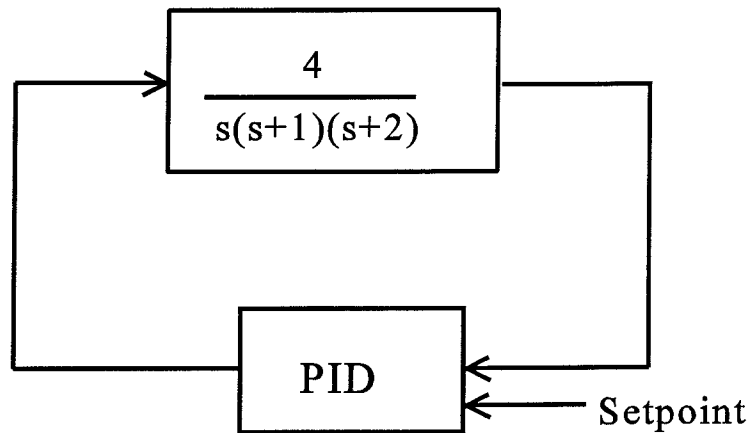


Fig. 59. Basic Block Diagram of a Demonstration PID Control Loop

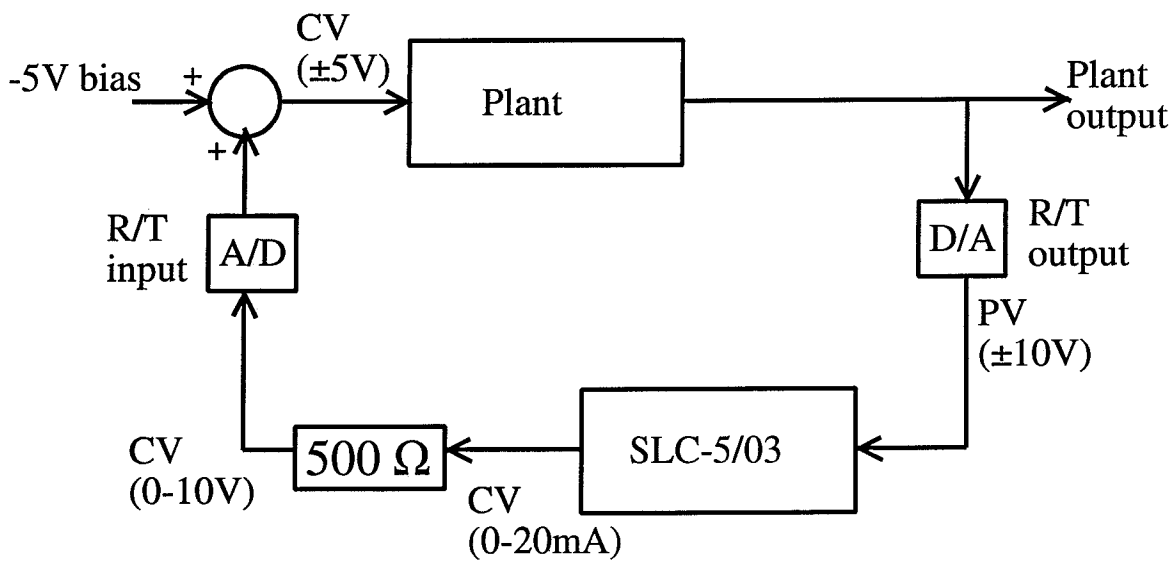


Fig. 60 PID Control Loop Simulation with SLC-5/03 Hardware in the Loop

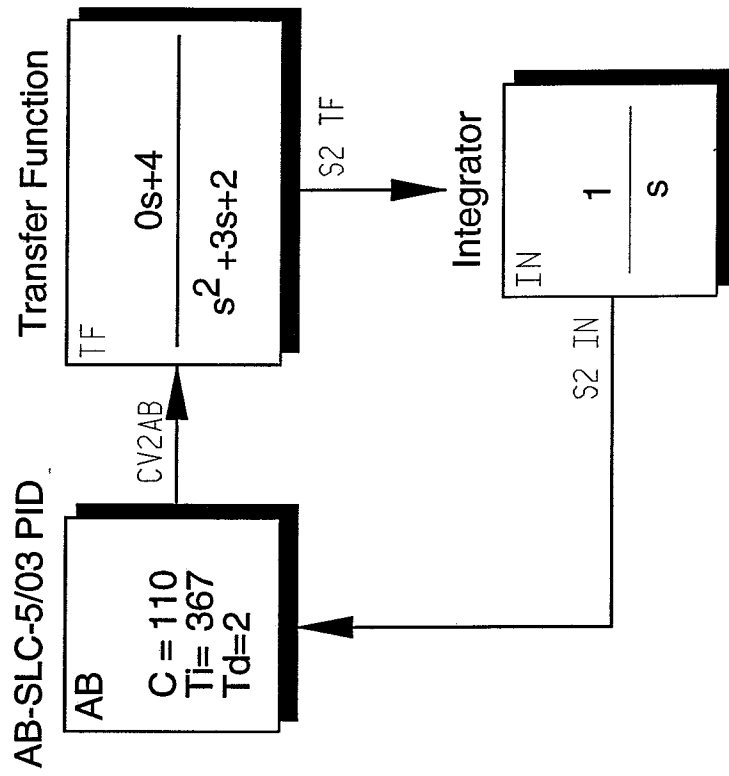


Fig. 61. EASY/ROCE TS Block Diagram for Demonstration System

In order to compare the outputs of the system with the SLC-5 hardware in the loop and the system which is entirely simulated in the EASY/ROCETS environment, the following values were used in both systems:

SP = 819.2	(PID form of setpoint corresponding to 1 V)
RG = 1	(use the extended range, which divides C and TI by 100)
C = 110	($K_c = 1.1$, then multiply by 100)
TI = 367	($T_i = 3.67$, then multiply by 100)
TD = 2	($T_d = 0.02$, then multiply by 100)
TAU = 0.01	(scan time of 10 msec.)
SC = 1	(scale the setpoint and the deadband)
AP = 819.2	(scaling rate)
APO = 0	(scaling offset)
BI = 8192	(bias the output by $\frac{1}{2}$ full scale)
DB = 0	(no deadband for this example)

Figure 62 shows the output of the hardware in the loop version of the system with the results of the EASY/ROCETS simulation. There are obviously some differences in the two responses. These differences are shown in Table 11, along with the relative error for each parameter. These differences are attributable to the lack of a detailed knowledge of the actual equation used in calculation of the PID control variable. The detailed algorithm used in the hardware would allow precise calculation of the responses, including quantization effects associated with the integer arithmetic employed. However, as has been mentioned previously, the details of these calculations are proprietary information, and are not available at this time.

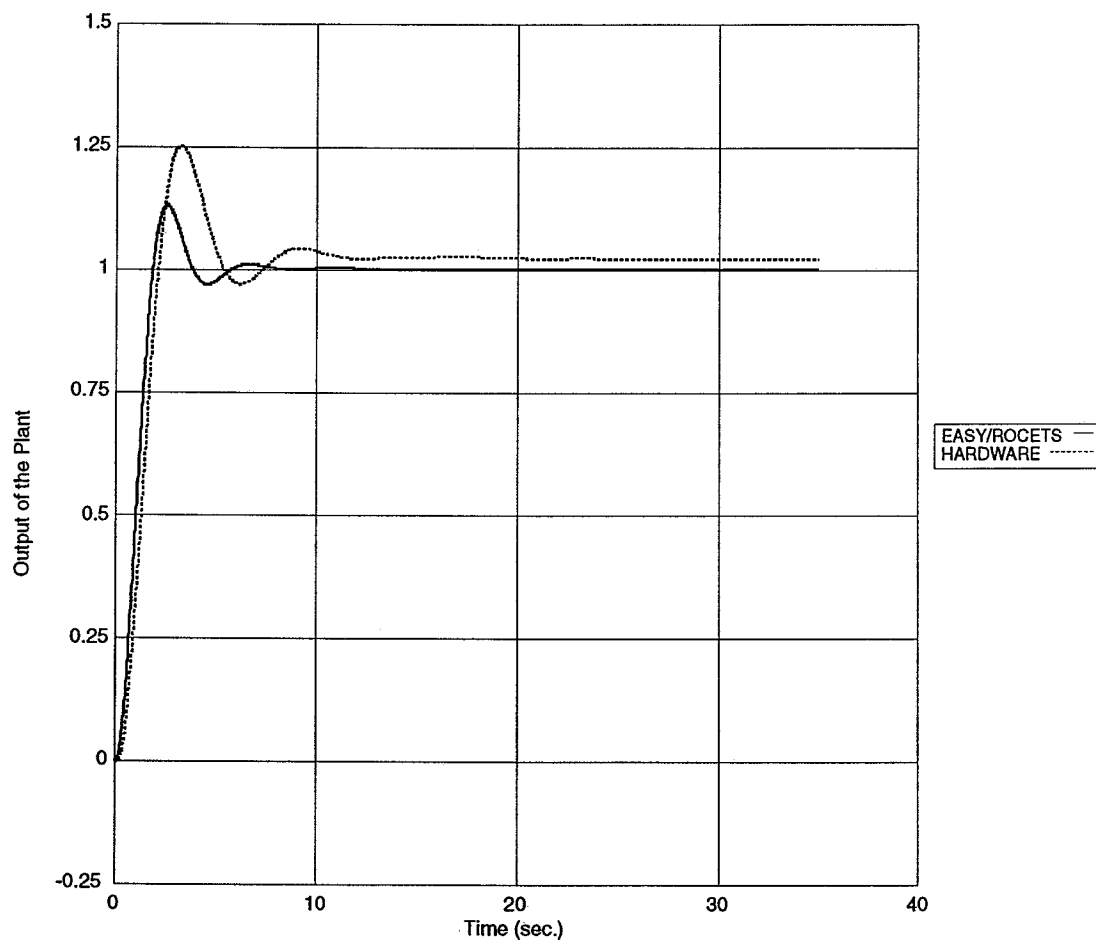


Fig. 62. Demonstration System Response for Hardware Testing and EASY/ROCETS Simulation (No Deadband)

Parameter	Hardware Value	Simulated Value	% Error
Time-to-Peak	3.220	2.578	19.9%
Peak Value	1.254	1.195	4.7%
Steady-State Value	1.022	1.003	1.9%

Table 11. Comparison of the Hardware and Simulated Responses for No Deadband.

Next, the system was simulated with exactly the same values as before, except that a deadband was inserted with magnitude 100. The results are shown in Figure 63, and the results are summarized in Table 12. The responses again are similar, but have some significant differences. It is believed that the hardware response has more of an oscillatory response due to the fact that the peak value was higher and the deadband played a more significant role in the response. If the amplitude scaling is adjusted so that there is a smaller peak value, the response does not go as far out of the deadband, and does not develop the larger control variable that results from the response shown.

Parameter	Hardware Value	Simulated Value	% Error
Time-to-Peak	3.125	2.578	17.5%
Peak Value	1.296	1.157	10.7%
Steady-State Value	1.033	1.004	2.8%

Table 12. Comparison of the Hardware and Simulated Responses with Deadband

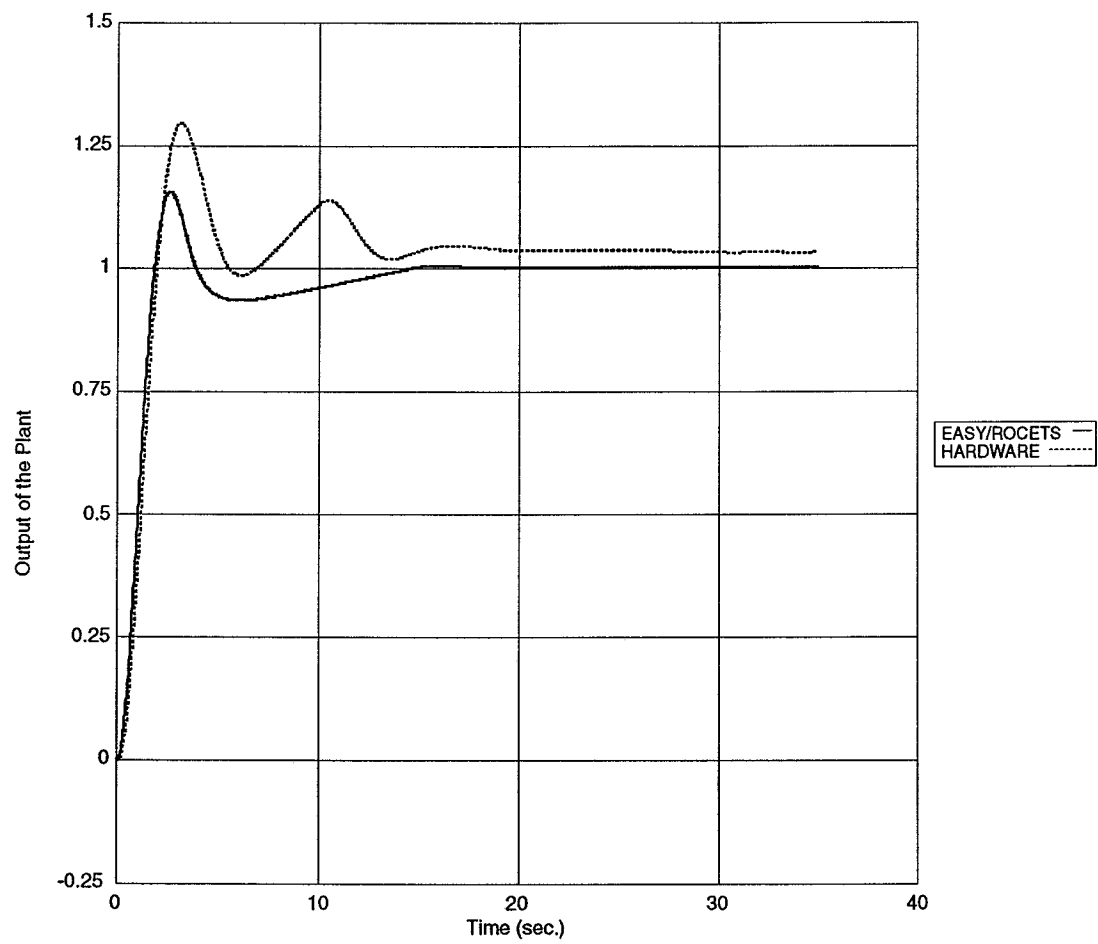


Fig. 63 Demonstration System Response for Hardware Testing and EASY/ROCETS Simulation (Deadband of 100)

Conclusions

It is evident from the previous simulations that the EASY/ROCETS module AB gives a reasonable approximation to the response of the actual Allen-Bradley SLC-5/03 hardware. The differences are accounted for by the lack of detailed knowledge of the hardware implementation of the PID control law. However, the general form of the response is reproduced by the simulation, and can still provide simulations which can be used in the design of PID control laws for use in systems which can be modeled by EASY/ROCETS. Once the PID controller is designed and simulated in EASY/ROCETS, the results can be used as a baseline from which to fine-tune the details of the controller. This will significantly reduce the number of iterations required to choose the controller constants, with resulting savings in both time and cost.

Valve Module with Delay (Component XD)

The valve module with delay is adapted from one developed by Steve Poulton to model compressible fluid flow through a valve. The valve actuation has been modified to include a delay term to give a more realistic model of response times. The following section shows a brief derivation of the code and discusses the usage of the module in the EASY/ROCETS environment.

Derivation and Use

A preliminary analysis of the time delays associated with changes in valve stem position has been made by Follett and Taylor (1996). The delay time was modeled by a lag function, which gives an exponentially varying response time in response to a step-type input. Figure 64 shows a schematic of this implementation, and Figure 65 shows response curves comparing the command given to the valve and the simulated response of the valve. Note that in the model shown, the fully open flow coefficient is the parameter being varied rather than the percent open coefficient. By connecting the control input to the variable representing the percent open, the valvestem position could be directly controlled. As is seen in the resulting plot, a much more realistic response curve is obtained by this technique. The variable which controls the speed of the response is the time coefficient in the denominator of the lag function.

The details of the operation of this new module are clearly evident when the multiplication by the lag function is considered. A lag function has a general form of

$$G_{\text{lag}} = \frac{1}{\tau s + 1} \quad (13)$$

and can be represented in the time domain by a multiplication by a time delay of

$$t_{\text{delay}} = e^{-t/\tau} \quad (14)$$

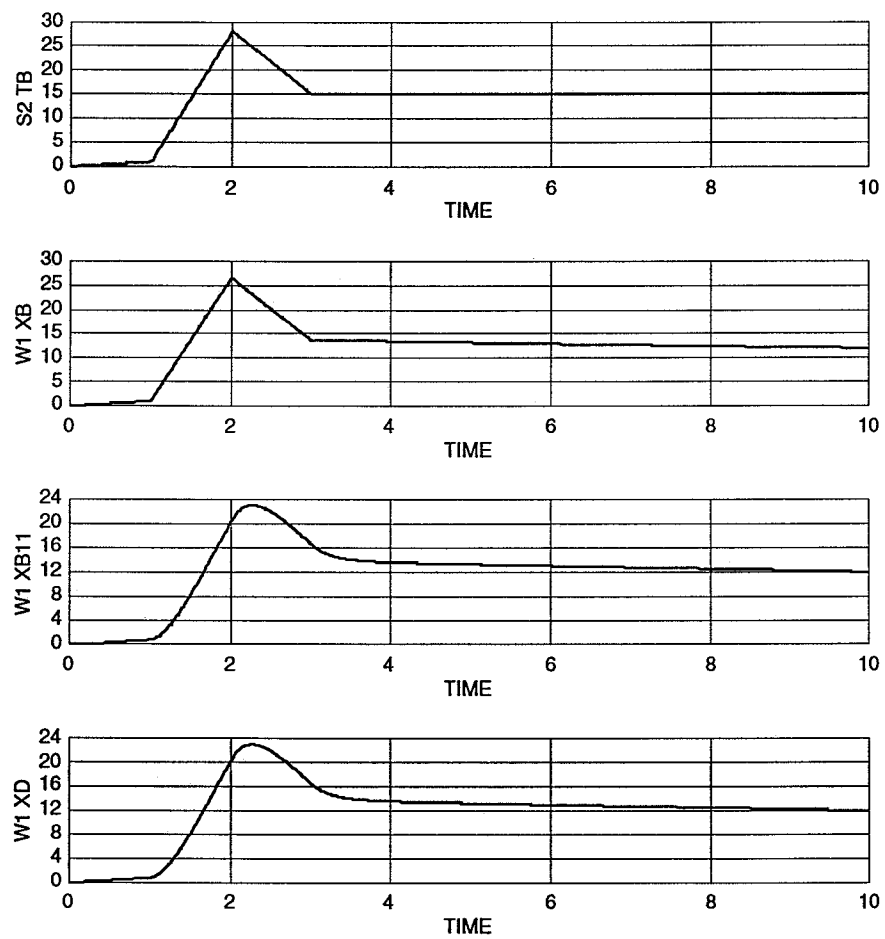


Fig. 65 Responses for Demonstration Case for Valve Delay Module

with the variable τ representing the time constant of the lag. In a period of time equal to four time constants, the response has traveled through 98% of the change that it will make. So, a valve with this lag function built in will change by 98% of the commanded change within this time period.

The new valve module accepts as an input the maximum actuation time specification for the valve. Internally, this number is divided by four to obtain the time constant to be used in the module. Additionally, the default value of zero can be used to obtain ideal, or instantaneous responses. Also, the user can specify whether the delay is to be applied to the fully open flow coefficient or the valve percent open. Either one of these parameters can actually be used as the control variable for the valve, and therefore must be able to be delayed by the valve actuation time. A copy of the valve icon for the new delay valve is given in Figure 66.

Valve 98 with time delay

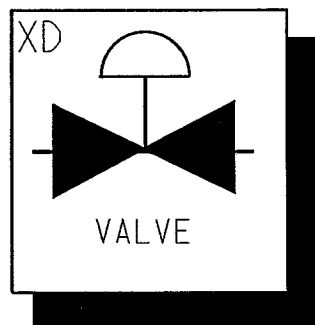


Fig. 66. EASY/ROCETS Component Icon for Valve with Delay

SECTION 5

CONCLUSIONS AND RECOMMENDED FUTURE WORK

Several advances in the state of the EASY/ROCETS dynamic fluid flow simulation package have been presented in this report. Incorporating the complete NIST-12 property database into the EASY/ROCETS library has significantly advanced the thermodynamic property capabilities of the system. This involves a complete new library "nr-library," which has been redesigned for more natural data entry for the initial conditions. New modules have been developed for the Allen-Bradley SLC 5/03 programmable logic controller PID system and for a valve with time delay actuation. Models of the E-1 Test Stand subsystems have been updated to the most recent design data. Using the new nr-library and the revised er-library, these updated models now work for all pressure levels.

EASY/ROCETS has reached a level of maturity such that it is useful for performing simulations of flow systems in support of facility design, test scenario development, facility modification and facility control. Further augmentation of the package should incorporate feedback from system engineers at SSC. The future usefulness of EASY/ROCETS depends to a great extent on the perceptions of the individual engineers regarding ease of use and validation of the output. Once system models are used on a regular basis, clear plans can be made for specific upgrades or changes to the software.

Some specific recommendations for future work can be made at this time. EASY/ROCETS should be worked heavily to develop control system designs and to work on the initial tuning of the PID controllers. The package has not been used in this

mode much to date and both the present researchers (Follett and Taylor) and our colleagues at SSC need to exercise the package in this mode to determine how modeling can help with the control system and what modeling strategies need to be worked out, to tune controllers for example. Also, more systems at SSC should be modeled in this system.

Validations should still be performed by comparison with actual facility data accumulated during test runs. As tests are made on systems that have been modeled, the actual results should regularly be compared with the model predictions, and the comparison used to increase the fidelity of the models. Detailed analysis by simulation experts should occur on a regular basis.

SECTION 6

REFERENCES

DeQuay, L., 1996, Personal Communication, NASA John C. Stennis Space Center, MS.

DeQuay, L., 1997, Personal Communication, NASA John C. Stennis Space Center, MS.

Follett, R. F. and Taylor, R. P., 1996, "CTF Dynamic Fluid Flow Modeling Program," Final Report NAS13-564, Delivery Order, 83, NASA John C. Stennis Space Center, MS.

McCarty, R. D., 1975, "Hydrogen Technological Survey—Thermophysical Properties," NASA SP-3089.

Phillips, C. L. and Harbor, R. D., 1996, Feedback Control Systems, 3rd edition, Prentice Hall, New Jersey.

Taylor, R. P., and Follett, R. F., 1994, "Technical Support for Dynamic Fluid Flow Simulation Capabilities for Ground Test Facility Modeling," Final Report NAS13-564, Delivery Order 16, NASA John C. Stennis Space Center, MS.

APPENDIX I

HANDWRITTEN NOTES ON MODEL DEVELOPMENT

HPH2.....	119
HPO2.....	135
LPO2.....	143

High-Pressure Hydrogen System

I Model of the Piping from Bottles to Split GØ - MPS

From G4 to G4 there are 3 parallel lines with the following lengths & diameters

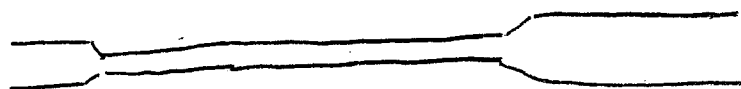
$\frac{D}{\text{dia}}$	$\frac{L}{\text{act length}}$	$\frac{EL}{\text{Eq. length}}$
4"	2.438'	2.438'
3.651"	8.271'	8.271'
3.651"	8.392'	14.782'
4.751"	5.335'	11.272'

When minor losses are accounted for in equivalent lengths.

The first computation is the actual volume

$$Vol = 3 \sum \frac{\pi}{4} D^4 L = \underline{10,788 \text{ in}^3}$$

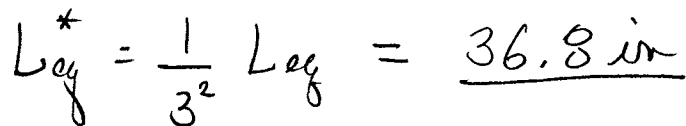
The next calculation is the reduction of each leg to an equivalent length at a single reference diameter. Here the reference diameter is taken to be 3.651" since most of the head loss will be in this leg



$$L_{eq} = \sum EL \left(\frac{3.651}{D} \right)^5 = \underline{331.4 \text{ in}}$$

13-792 500 SHEETS, FILLER 5 SQUARE
42-381 50 SHEETS EYE-EASE® 5 SQUARE
42-382 100 SHEETS EYE-EASE® 5 SQUARE
42-389 200 SHEETS EYE-EASE® 5 SQUARE
42-392 100 RECYCLED WHITE 5 SQUARE
42-399 200 RECYCLED WHITE 5 SQUARE

Made in U.S.A.



National® Brand

D	L	EL
4.75)	5.097'	5.097'
3.65)	1.042'	1.042'
3.65)	1.042'	1.042'
4.75)	0.208'	0.208'
4.75)	26.566'	37.756'
4.75)	28.586'	28.583'
3.65)	30.479'	30.479'

$$Vol = \sum \frac{\pi}{4} D^2 L = 16,950 \text{ in}^3$$
$$L_{eq} = \sum EL \left(\frac{3.65}{D} \right)^2 = 621.16''$$

The K-factors of the valves are computed
for $d = 3.651''$ & $C_v = 700$

$$K = \left(\frac{29.9 d^4}{C_v} \right)^2 = 0.324$$

For two valves in series

$$K_{TOT} = 2K = \underline{0.648}$$

Summary

$$\begin{aligned} L_{eq} &= 36.8 + 621.16 = \underline{657.96''} \\ D_{eq} &= \underline{3.651''} \\ K &= \underline{0.648} \end{aligned}$$

These are the dimensions for module PA in
Fig 2.

$$\begin{aligned} Vol. &= 10,788 \text{ in}^3 + 16,950 \text{ in}^3 \\ Vol &= \underline{27,738 \text{ in}^3} \end{aligned}$$

This volume will be included in module VL
in Figure 2.

II Model of the Tank Pressurization Line MPS - PH

This system is modeled using modules PA11, VO, and XB in Fig 2. All of the piping and the two shutoff valves are modeled in PA11. The two flow control valves are modeled in XB. The two venturi valves can be modeled as a single valve since CV's add for parallel valves.

The pipe sizes are

D	L	EL
3.651"	63.035'	87.03'
4.018"	9.379'	13.8'
5.282"	27.649'	65.25'

The actual volume

$$Vol = \sum \frac{\pi}{4} D^2 L = \underline{16,640 \text{ in}^3}$$

The equivalent length of 3.651" pipe

$$L_{eq} = \sum EL \left(\frac{3.651}{D} \right)^5 = 1267 \text{ in}$$

The K-factors for the valves based on
 $d = 3.651$ when $C_v = 230 \pm 700$

$$K_{\text{Valv}} = \left(\frac{29.9 \text{ ft}^2}{C_v} \right)^2 = 3 + 0.324 = \underline{3.324}$$

Since all of the kinetic energy is lost in
the diffuser

$$K_{\text{diff}} = \underline{1.0}$$

Summary

$$\begin{aligned} L_{\text{eq}} &= \frac{1267 \text{ in}}{3.651 \text{ in}} \\ K_{\text{TOT}} &= 4.324 \end{aligned}$$

These are the dimensions of module PA11

The volume is split between V1 (8320)
and V0 (8320)

III Model of the mixer gas system MPS - MGA

First the parallel legs MGB - MGA are considered. Each leg has the following dimensions

$$\begin{array}{l} D \\ 4.6'' \\ 2.3'' \end{array}$$

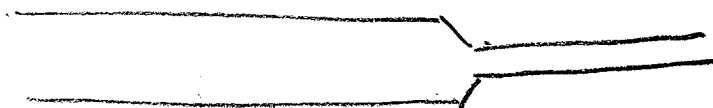
$$\begin{array}{l} L \\ 5.164' \\ 1.646 \end{array}$$

$$\begin{array}{l} EL \\ 10.74' \\ 1.646' \end{array}$$

Actual volume

$$Vol = 2 \sum \frac{\pi}{4} D^2 L = \underline{2224 \text{ in}}$$

The equivalent length of 3.651" pipe for each leg



$$L_{eq} = \sum EL \left(\frac{3.651}{D} \right)^5 = \underline{239.7 \text{ in}}$$

The equivalent length of the parallel legs



$$L_{eq}^* = \frac{1}{2^2} L_{eq} = \underline{59.9 \text{ in}}$$

pipe from MPS - MAB

D	L	EL
3.651"	18.019'	40.223'
5.99	2.252'	2.252'

Actual Volume

$$Vol = \sum \frac{\pi}{4} D^2 L = \underline{3025 \text{ in}^3}$$

Equivalent length of 3.651" pipe

$$L_{eq} = \sum EL \left(\frac{3.651}{D} \right)^5 = \underline{484.9 \text{ in}}$$

K-factor for the valve $C_v = 230$ based on 3.651" pipe

$$K = \left(\frac{29.9 d^2}{C_v} \right)^2 = \underline{3}$$

K-factor for the venturi - estimate = 0.3

K-factor for Exit - K = 1.0

Summary

$$L_{eq} = 484.9 + 59.9 = \underline{544.8 \text{ in}}$$

$$d_{eq} = \underline{3.651 \text{ in}}$$

$$K_{TOT} = \underline{4.3}$$

These are the dimensions of PA12

The total volume (5249 m^3) is split between VI (2624.5) and VØ11 (2624.5)

IV Model for Liquid System AO-C11

The pipe is lumped together and is modeled as one $4.058''$ pipe.

D	L	EL
8.5"	7.492'	7.492'
5.474"	1.083'	1.083'
5.282"	11.384'	15.105'
5.282"	2.167'	2.167'
4.058"	10.115'	10.115'
4.058"	48.275'	59.991'
3.134"	10.136'	13.001'
3.134"	2.042'	2.042'
3.134"	3.76'	3.76'
3.134"	1.896'	1.896'

Compute actual volume

$$Vol = \sum \frac{\pi}{4} D^2 L = \underline{19700 \text{ m}^3}$$

This volume is placed in module VØ12.

The equivalent length of $4.058''$ pipe

$$L_{eq} = \sum EL \left(\frac{4.058}{D} \right)^4 = \underline{1806 \text{ m}}$$

K factor for the valve $C_v = 1600$

$$K_{valv} = \left(\frac{29.9 (4.058)^2}{1600} \right)^2 = 0.095$$

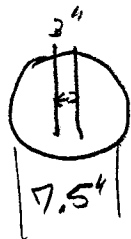
K-factor for reducer 8.5" - 5.479"
assume 45° cone

$$K_{8.5} = \frac{0.8 \sin 45^\circ \left(1 - \left(\frac{5.479}{8.5}\right)^2\right)}{\left(\frac{5.479}{8.5}\right)^4} = 1.915$$

Based on 4.058" pipe

$$K_{4.058} = K_{8.5} \left(\frac{4.058}{8.5}\right)^4 = \underline{0.099}$$

Mixer K-factor



$$\beta^2 = \frac{A_1}{A_2} = \frac{3 \times 7.465}{\frac{\pi}{4} 7.465^2} = 0.511$$

Use Crane Manual formula for a gate valve

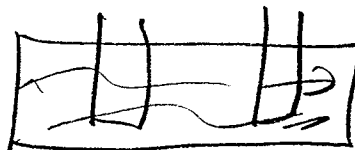
$$K_2 = \frac{8(0.014) + \frac{1}{2} \sqrt{\sin 90^\circ (1 - \beta^2)} + (1 - \beta^2)^2}{\beta^4}$$

$$K_2 = 2.28$$

Based on 4.058"

$$K = 2.28 \left(\frac{4.058}{7.465}\right)^4 = 0.199$$

2-restrictions



$$K = 0.398$$

Assume 10-1 density reduction
in mixer

$$\underline{K_{mix} = 3.98}$$

Summary

The dimensions for PA13 are

$$L_{eq} = 1806 \text{ in}$$

$$D_{eq} = 4.058 \text{ in}$$

$$K_{Tot} = 3.98 + 0.099 = \underline{4.079}$$

The volume of V012 is 19,700 in³

The parallel flow control valves are modeled
using module XA.

V Mixer volume C11 - E14

C11 - B12

$$4' 10\frac{1}{2}" = 58.5" \text{ long}$$

$$dia = 7.465"$$

$$Vol = \frac{\pi}{4} 7.465^2 \times 58.5 = 2560 \text{ in}^3$$

E12 - E14

$$Vol = 5.208 \times 12 \times \frac{\pi}{4} \times 5.99^2 = 1761 \text{ in}^3$$

$$Vol_{Tot} = \underline{4321} \text{ in}^3 \quad \text{This goes in V11}$$

VI Mixer to Test cell #2 side E14-H1

VIII Volume Summary

$$V1 = \underbrace{27,738}_{G\phi-MPS} + \underbrace{8320}_{MPS-P11} + \underbrace{2624.5}_{MPS-MGA} = \underline{38682} \text{ in}^3$$

$$V\phi = \underbrace{8320}_{MPS-P11} \text{ in}^3$$

$$V\phi11 = \underbrace{2624.5}_{MPS-MGA} \text{ in}^3$$

$$V11 = \underbrace{4321}_{C11-E14} \text{ in}^3$$

$$V012 = \underbrace{19,700}_{A0-C11} \text{ in}^3$$

$$V013 = \underbrace{12,416}_{(E14-H1) \text{ Test cell 2}} \text{ in}^3$$

$$V014 = \underbrace{2538}_{(E14-H1) \text{ Test cell 3}} \text{ in}^3$$

13-782 500 SHEETS, FILLER 5 SQUARE
42-381 50 SHEETS, EYE-GLASS 5 SQUARE
42-382 100 SHEETS, EYE-GLASS 5 SQUARE
42-383 200 SHEETS, EYE-GLASS 5 SQUARE
42-384 100 SHEETS, EYE-GLASS 5 SQUARE
42-385 100 RECYCLED WHITE 5 SQUARE
42-386 200 RECYCLED WHITE 5 SQUARE
Made in U.S.A.



IX PIPE SUMMARY

	L	D	K	E	F1	F2
PA	658in	3.65in	6.648	0.0018in	—	—
PA11	1267in	3.65in	4.324	0.0018in	—	—
PA12	545in	3.65in	4.3	0.0018in	—	—
PA13	1806in	4.05in	4.079	0.0018in	—	—
PS	687in	5.99in	6.2	0.0018in	0.9863	0
PS11	893in	2.3in	6.1	0.0018in	0.9863	0

X Flow Control Value Summary

	CV ₁	CV ₂	X _{L1}	X _{L2}
XA	115	22	—	—
XA11	470	—	—	—
XA12	70	—	—	—
XB	110	28	0.54	0.54
XB11	230	18	0.54	0.54

15 782 500 SHEETS FULLER SQUARE
 42-281 50 SHEETS EYE-EASE SQUARE
 42-382 100 SHEETS EYE-EASE SQUARE
 42-383 200 SHEETS EYE-EASE SQUARE
 42-384 400 SHEETS EYE-EASE SQUARE
 42-385 800 SHEETS EYE-EASE SQUARE
 42-386 200 RECYCLED WHITE SQUARE
 42-387 200 RECYCLED WHITE SQUARE
 Made in U.S.A.



High-Pressure Oxygen System

13-782 500 SHEETS, FILLER 5 SQUARE
42-381 50 SHEETS EYE-EASE® 5 SQUARE
42-382 100 SHEETS EYE-EASE® 5 SQUARE
42-389 200 SHEETS EYE-EASE® 5 SQUARE
42-392 100 RECYCLED WHITE 5 SQUARE
42-399 200 RECYCLED WHITE 5 SQUARE

Made in U.S.A.



National® Brand

The minor losses are accounted for in the eq. lengths

$$V_0 = 4 \sum \frac{\pi}{4} D^2 L = 10,230 \text{ m}^3$$
$$L_{eq} = \frac{1}{4^2} \sum EL \left(\frac{4.751}{D} \right)^5 = \underline{115.2 \text{ in}}$$

136

From G 3 to G 17 we have the following data

D	L	EL
3.651 ^u	18.5'	18.5'
3.651 ^u	6.083'	6.083'
3.651 ^u	0.542'	6.543'
4.751 ^u	12.625'	12.625'
4.751 ^u	26.566'	37.756'
4.751 ^u	47.667'	47.667'
4.751 ^u	48.666'	57.441'
4.751 ^u	8.362'	11.543'
3.651 ^u	0.542'	6.542'
3.651 ^u	18.146'	22.409'
3.651 ^u	5.337'	9.6'
3.651'	8.212'	12.475'

$$V_{\text{ol}} = \int \frac{\pi}{4} D^2 L = \underline{37816 \text{ m}^3}$$

$$L_{eq} = \sum EI \left(\frac{4.751}{D} \right)^5 = \underline{5146 \text{ in}}$$

From G_{17} to G_{22} we have 2 parallel pipes

D	L	EL
3.651"	4.833'	22.984'
3.651"	8.546'	12.809'
3.651"	1.720'	4.193'
3.651"	0.433'	0.433'
3.5"	12.156'	18.317'

$$V_{\text{ol}} = 2 \sum \frac{\pi}{4} \bar{D}_L^2 = 6,612 \text{ m}^3$$

$$L_{eq} = \frac{1}{2^2} \sum EI \left(\frac{4.751}{D} \right)^2 = 703$$

Between the bottles and the flow control valves there are 3 valves with C_v 's of 700, 700, and 230. These are converted into K-factor equivalents for 4.751-in pipe

$$K = 2 \left(\frac{29.9 (4.751)^2}{700} \right)^2 + \left(\frac{29.9 (4.751)^2}{230} \right)^2$$

$$K = 10.47$$

The totals of the N_2 pressurization piping is

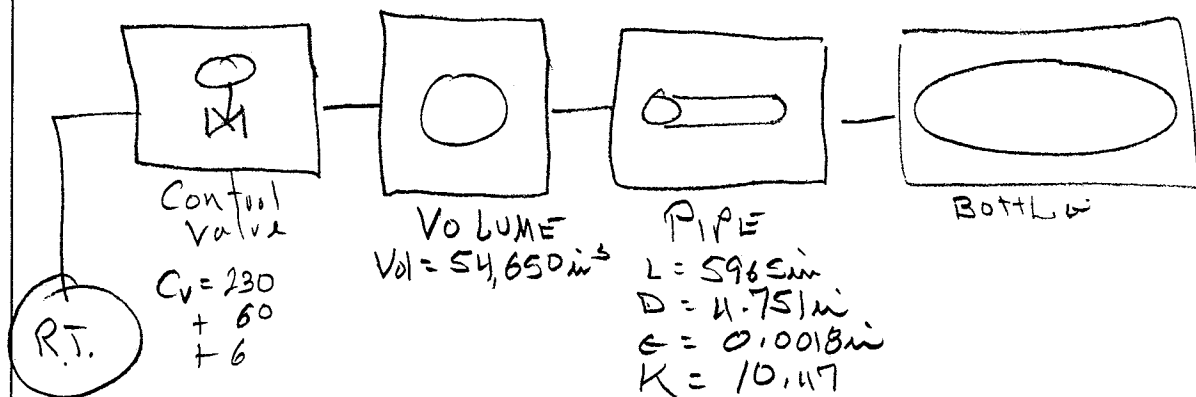
$$Vol = 54,650 \text{ in}^3$$

$$L = 5913 \text{ in}$$

$$D = 4.751$$

$$K_{eq} = 10.47$$

The EASY/ROLETS MODEL FOR THE N_2 system is



At A7 the line splits. 2 parallel 5.282" ϕ lines head toward Test Cell 3 and one line heads toward Test cell 2.

For the test-cell 2 side

$$\begin{array}{c} D \\ 2.1144 \\ 2.3 \end{array}$$

$$\begin{array}{c} \downarrow \\ 2.3 \end{array}$$

$$\begin{array}{c} L \\ 5.365' \\ 3.740' \\ 11.462' \\ 7.75' \\ 8.962' \\ 6.438' \\ 9.75' \\ 9.25' \end{array}$$

$$\begin{array}{c} E \\ 15.907' \\ 3.74' \\ 15.758' \\ 7.75' \\ 13.758' \\ 6.438' \\ 9.75' \\ 9.25' \end{array}$$

$$Vol = \frac{\pi}{4} \sum D^2 L = 2786 \text{ in}^3$$

$$L_{eq} = \sum E L \left(\frac{2.3}{D} \right)^5 = 1003 \text{ in}$$

Minor losses come from a valve with $C_v = 70$, a expansion - from 2.144 to 2.3", a venturi $K = 0.7$ estimated and an exit loss $K = 1.0$

$$K_{eq} = \left(\frac{29.9 \cdot 2.3^2}{70} \right)^2 + 0.5 \frac{(1 - B^4)}{B^4} + 1 + 0.7$$

$$K_{eq} = 6.893$$

For Test Cell 3, we have 2 parallel lines
from A7 to C12

D	L	EL
5.282 ⁴	4.01	44.102'
	1.472'	27.399'
	0.583'	0.583'
	3.642'	11.065'
	4.142'	11.565'
	23.925'	27.637'
	4.142'	11.565'
	3.142'	10.065'
	0.25'	0.25'
	1.25'	1.25'
	8.0'	8.0'
5.282 ⁴	4.31	8.012'

$$Vol = \sum \frac{\pi}{4} D^2 L = 15,470 \text{ in}^3$$

$$L_{eq} = 1938 \text{ in}$$

For minor losses there are 2 elbows $C_v = 478$

$$K_{eq} = 2 \cdot \left(\frac{29.9 \times 5.282^2}{478} \right)^2 = 6.3$$

From C12 to F1 there is again one line

D	L	EL
5.282	1.167'	27.138
	0.583'	0.583
	10.154'	13.866
	4.071'	7.782
	7.675'	11.387
	0.583'	0.583
	2.154'	5.866

$$Vol = \frac{\pi}{4} \sum D^2 L = 6,938 \text{ in}^3$$

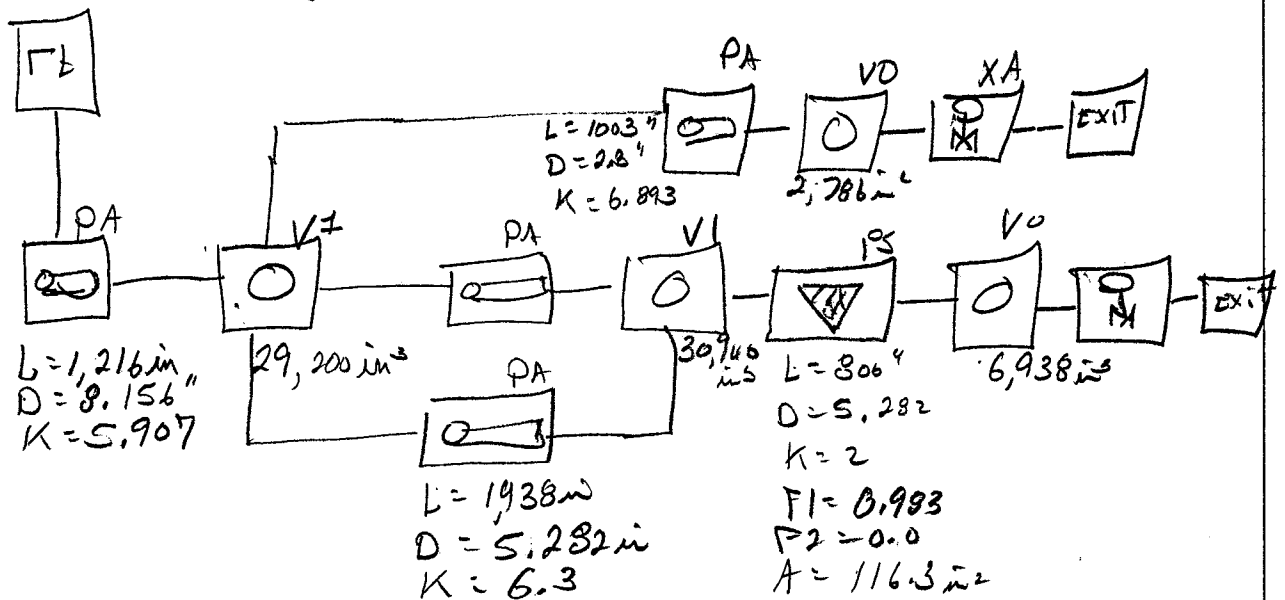
$$L_{eq} = 806 \text{ in}$$

13-782 500 SHEETS FILLER 5 SQUARE
42-381 50 SHEETS RELEASE 5 SQUARE
42-382 100 SHEETS RELEASE 5 SQUARE
42-383 100 SHEETS RELEASE 5 SQUARE
42-384 100 SHEETS RELEASE 5 SQUARE
42-385 100 SHEETS RELEASE 5 SQUARE
42-386 100 SHEETS RELEASE 5 SQUARE
42-387 100 SHEETS RELEASE 5 SQUARE
42-388 100 SHEETS RELEASE 5 SQUARE
42-389 100 SHEETS RELEASE 5 SQUARE
42-390 100 SHEETS RELEASE 5 SQUARE
42-391 100 SHEETS RELEASE 5 SQUARE
42-392 100 SHEETS RELEASE 5 SQUARE
42-393 100 SHEETS RELEASE 5 SQUARE
42-394 100 SHEETS RELEASE 5 SQUARE
42-395 100 SHEETS RELEASE 5 SQUARE
42-396 100 SHEETS RELEASE 5 SQUARE
42-397 100 SHEETS RELEASE 5 SQUARE
42-398 100 SHEETS RELEASE 5 SQUARE
42-399 100 SHEETS RELEASE 5 SQUARE
42-400 100 SHEETS RELEASE 5 SQUARE
National Brand
Made in U.S.A.

Minor loss coefficients are for the exit loss
and Turbine meter

$$K = 1 + 1 = 2$$

This line also contain a 246 μ Screen with
an area of 116.37 in²



Low-Pressure Oxygen System

I. N₂ lines are unchanged from previous model

II LIQUID O₂ SYSTEM

A. A ϕ to A1E

12" ϕ pipe with $L_{eq} = 50.5'$ and

$L_{straight} = 30.135' \Rightarrow \text{Volume} = 43,531 \text{ in}^3$

a) Model the flow meter as a minor-loss factor with $K = 0.8$.

b) Model the screen ($\frac{5}{32}"$) as an orifice with $\beta = 0.8 \Rightarrow K = 1.8$

Use a vlv pipe with

$$C_v = 7066$$

$$L = 606"$$

$$K = 2.6$$

$$D = 12$$

$$e = 0.0018$$

B. A1 to Test cell 3

12" ϕ pipe $L_{eq} = 50.6'$, $L_{straight} = 27.28'$
 $\Rightarrow \text{Vol} = 37,022 \text{ in}^3$

Model the first half as a vlv pipe and the second half as a screen.

Vlv pipe

$$C_v = 7066$$

$$L = 303.4"$$

$$K = 0$$

$$D = 12$$

$$e = 0.0018$$

Screen

$$L = 303.4$$

$$K = 0$$

$$D = 12$$

$$e = 0.0018$$

$$A = 250$$

$$F1 = 0.9863$$

$$F2 = 0$$

C. A1 to TEST CELL 2

12" d pipe $L_{eq} = 107.4'$, $L_{st} = 57.25'$

$$V_0 = 77,699 \text{ in}^3$$

Model the first half with a vlv pip and the second with a screen

Vlv p: p

$$L = 644$$
$$C_v = 7060$$
$$K \approx 0$$
$$D = 12$$
$$E = 6.0018$$

Screen

$$L = 644$$
$$D_r = i2$$
$$K \geq 0$$

$\epsilon = 0.0018$

$$A = 250 \text{ m}^2$$
$$F_1 = 0.9863$$
$$F_2 = 0.0$$

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

June 30, 1998

3. REPORT TYPE AND DATES COVERED

Final Contractor Report

4. TITLE AND SUBTITLE

E-1 DYNAMIC FLUID-FLOW MODEL UPDATE

5. FUNDING NUMBERS

NAS 13-564 DO 143

6. AUTHOR(S)

Randolph F. Follett, Electrical and Computer Engineering Department
Robert P. Taylor, Mechanical Engineering Department

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Mississippi State University
P.O. Drawer ME
Mississippi State, MS 39762

8. PERFORMING ORGANIZATION
REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NASA
John C. Stennis Space Center
Building 1100, Code VA32
Stennis Space Center, MS 39529-6000

10. SPONSORING/MONITORING
AGENCY REPORT NUMBER

SE-1999-03-00014-SSC

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Publicly Available

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Computer models of dynamic fluid flow simulation are needed to predict the pressures, temperatures, flowrates, etc. in present and future testing operations at the NASA John C. Stennis Space Center. Such simulations are used in facility design, test scenario development, facility modifications, and facility control. The ROCETS package, which was initially developed by Pratt & Whitney for NASA MSFC, and EASY 5x, a commercial package developed by the Boeing Co., are the two major components that comprise the EASY/ROCETS dynamic fluid flow simulation package developed by MSU personnel for use by NASA/SSC. Additional code has been written to handle tasks specific to ground-test facility modeling such as gas bottles and pressurized liquid run tanks.

The present research incorporates the complete NIST-12 property database to enhance the thermodynamic property tables. E-1 Test Stand models for the low and high pressure run systems have been updated. New modules have been developed to model a PID programmable controller and for a valve with a time delay input.

14. SUBJECT TERMS

Computer Models, Fluid Flow Simulation

15. NUMBER OF PAGES

149

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION
OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION
OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

UNLIMITED